



UNIVERSIDAD CARLOS III DE MADRID
Departamento de Ingeniería Mecánica

PROYECTO FIN DE CARRERA

**ESTUDIO Y ANÁLISIS DE
FILTROS APLICADOS A
SEÑALES VIBRATORIAS DE
EJES FERROVIARIOS PARA LA
DETECCIÓN DE FALLOS**

TITULACIÓN

GRADO EN INGENIERÍA MECÁNICA

Autor

Eduardo Roldán Arroita

Tutora

María Jesús Gómez García

Directora

Cristina Castejón Sisamón

Leganés, Septiembre de 2012

AGRADECIMIENTOS

Quiero aprovechar estas líneas para dar las gracias a todas aquellas personas que me han acompañado en estos años de formación:

A mis profesores, que han contribuido en mi formación académica y personal, y en especial a mi tutora, María Jesús Gómez, por toda su paciencia y dedicación.

A mi familia, en especial a mis padres, por aguantar mis altibajos a lo largo de la carrera y apoyarme sin perder la fe en mí.

A mis mejores amigos: Garci, Kiko, Ortega, Chema, Poter y Pedro, porque con ellos he vivido muchos de los mejores momentos de mi vida y siempre han estado ahí cuando les necesitaba.

A mis compañeros, por acompañarme a lo largo de la increíble experiencia universitaria y aprender conmigo, ayudándome a sacar la carrera adelante.

A Paula, por su impulso para volverme responsable y madurar, que de no ser por ella no habría llegado hasta aquí.

RESUMEN

En el presente Proyecto de Fin de Carrera se han aplicado distintos filtros digitales a las señales de vibración de un eje de ferrocarril a escala, con el fin de determinar cuál nos lleva a un diagnóstico más adecuado.

Para ello, se ha adquirido un conjunto de señales de un banco de ensayos de ejes, midiendo las vibraciones de un eje sano y un eje con defecto. Las señales han sido procesadas, por separado, mediante el filtro Smooth, la Transformada de Hilbert y la Transformada Wavelet db6, que son los tipos de filtros a analizar.

Posteriormente, las señales han sido clasificadas por medio de una red neuronal artificial de reconocimiento de patrones, la cual es capaz de realizar un diagnóstico automático de la condición del sistema. A partir de la efectividad de esta clasificación se ha determinado la eficacia de cada filtrado.

El estudio realizado ha determinado que el filtrado elimina una gran parte del ruido aleatorio, mejorando mucho el resultado con respecto al obtenido a partir de la señal original. Aún así, las tasas de acierto siguen siendo insuficientes para la implementación al mantenimiento de ferrocarriles.

ABSTRACT

In the present Final Degree Project there have been applied different digital filters to vibration signals of a scaled railway axle in order to determine which one leads us to the most adequate diagnose.

To this end, a set of signals have been acquired from an axle test bench, measuring the vibrations of a healthy axle and a defected axle. The signals have been processed, separately, with Smooth filter, Hilbert Transform and db6 Wavelet Transform, which are the filters that are meant to be analyzed.

Subsequently, the signals have been classified by means of a pattern recognition artificial neural network, which is capable of performing an automatic diagnosis of the condition of the system. The efficiency of each filter has been determined from the effectiveness of the classification.

The study has determined that the filtering removes much of the random noise, greatly improving the result compared to the one obtained from the original signal. Anyway, success rates remain insufficient for the implementation to railway maintenance.

ÍNDICE

1. INTRODUCCIÓN	13
1.1 Antecedentes y motivación	15
1.2 Objetivo	16
1.3 Estructura del documento.....	17
2. MARCO TEÓRICO	19
2.1 Aplicación del procesamiento de la señal.....	19
2.2 El dominio de la frecuencia	20
2.3 La Transformada de Fourier	20
2.4 Transformada Discreta de Fourier (DFT).....	22
2.5 Transformada Rápida de Fourier (FFT)	23
2.6 Transformada Corta de Fourier (STFT)	23
2.7 Transformada Wavelet.....	24
2.8 Transformada Discreta Wavelet (DWT)	26
2.9 Transformada Wavelet Packets (WPT).....	27
2.10 Transformada de Hilbert.....	28
2.11 Sistemas clasificadores inteligentes.....	30
2.12 Redes neuronales artificiales (ANN)	31
3. SISTEMA EXPERIMENTAL DE MEDIDA.....	35
3.1 Introducción	35
3.2 Componentes del equipo de ensayo	36
3.3 Adquisición de datos	43

4. PROCESAMIENTO DE LAS SEÑALES Y ANÁLISIS PRELIMINARES	45
4.1 Introducción	45
4.2 Selección de señales a estudiar	45
4.3 Señal original	48
4.3.1 Señal temporal original	48
4.3.2 Señal en el dominio de la frecuencia (FFT)	51
4.4 Filtro Smooth	54
4.5 Filtrado mediante la Transformada de Hilbert	60
4.6 Filtrado mediante la Transformada Wavelet (TW)	65
 5. REDES NEURONALES ARTIFICIALES	 71
5.1 Introducción	71
5.2 Paquetes de energía	73
5.3 Entrenamiento y test con todas las carpetas	74
5.3.1 Señales originales sin filtrar	74
5.3.2 Filtrado Smooth	80
5.3.3 Filtrado Hilbert	84
5.3.4 Filtrado Wavelet	88
5.4 Entrenamiento con unas carpetas y test con otras distintas	92
5.4.1 Señales originales sin filtrar	92
5.4.2 Filtrado Smooth	96
5.4.3 Filtrado Hilbert	99
5.4.4 Filtrado Wavelet	102
 6. CONCLUSIONES Y TRABAJOS FUTUROS	 105
6.1 Conclusiones	105
6.2 Trabajos futuros	108

7. REFERENCIAS.....	109
8. ANEXOS.....	111
Anexo I: Programa de Matlab® para filtrar y sacar las gráficas de una señal de vibración.....	111
Anexo II: Filtrado de todas las señales de una carpeta con un filtro determinado ..	114
Anexo III: Programa para obtener los histogramas de los valores medios, máximos y mínimos de las señales de una carpeta	117
Anexo IV: Programa para la obtención de los paquetes de energía Wavelet	118
Anexo V: Programa para crear la red neuronal a partir de todas las carpetas de señales (caso 1)	120
Anexo VI: Programa para crear la red neuronal a partir de unas señales y testear con otras diferentes (caso 2)	123

ÍNDICE DE TABLAS

Tabla 5.1: Comparación de las redes creadas con y sin filtrado de las señales.....	83
Tabla 6.1: Características de la red neuronal para cada tipo de filtrado (caso 1).....	105
Tabla 6.2: Características de la red neuronal para cada tipo de filtrado (caso 2).....	107

ÍNDICE DE FIGURAS

Figura 2.1: Excitación de sistemas para obtener sus datos.....	19
Figura 2.2: Análisis tiempo-frecuencia.....	22
Figura 2.3: Principales familias de Wavelets.....	25
Figura 2.4: Esquema del funcionamiento de las DWT.....	27
Figura 2.5: Esquema del funcionamiento de las WPT.....	28
Figura 2.6: Transformada de Hilbert de una onda cuadrada.....	30
Figura 2.7: Comparación entre red neuronal biológica y artificial.....	32
Figura 2.8: Red neuronal básica de tres capas.....	33
Figura 3.1: Banco de ensayos "Rotor Kit MFS".....	37
Figura 3.2: Motor eléctrico Marathon Electric.....	37
Figura 3.3: Rodamiento ER10K.....	38
Figura 3.4: Eje de ferrocarril a escala.....	39
Figura 3.5: Panel de control Delta.....	39
Figura 3.6: Tacómetro digital.....	40
Figura 3.7: Acelerómetro 4383.....	40
Figura 3.8: Amplificador 2693.....	41
Figura 3.9: Tarjeta de adquisición de datos KUSB-3100.....	42
Figura 3.10: Ordenador personal, tarjeta de adquisición de datos y amplificador.....	42
Figura 3.11: Cadena de medida.....	43
Figura 3.12: Btool.m.....	43
Figura 3.13: Representación gráfica Btool.m.....	44
Figura 4.1: Histograma de la media de las 100 señales de eje sano con poco ruido..	46
Figura 4.2: Histograma de los máximos de las 100 señales de eje sano con poco ruido.....	47
Figura 4.3: Histograma de los mínimos de las 100 señales de eje sano con poco ruido.....	48
Figura 4.4: Señales temporales de vibración de eje sano con poco ruido.....	49
Figura 4.5: Señales temporales de vibración de eje sano con mucho ruido.....	50
Figura 4.6: Señales temporales de vibración de eje con defecto al 100% del radio con poco ruido.....	50
Figura 4.7: Señales temporales de vibración de eje con defecto al 100% del radio con mucho ruido.....	51

Figura 4.8: FFT de las señales de eje sano con poco ruido.....	52
Figura 4.9: FFT de las señales de eje sano con mucho ruido.....	52
Figura 4.10: FFT de las señales de eje con defecto con poco ruido.....	53
Figura 4.11: FFT de las señales de eje con defecto con mucho ruido.....	54
Figura 4.12: Señales de eje sano con poco ruido procesadas mediante el filtro Smooth, en el dominio del tiempo y en el dominio de la frecuencia.....	55
Figura 4.13: Comparación de la señal temporal de vibración de eje sano con poco ruido sin filtrar y procesada mediante el filtro Smooth.....	56
Figura 4.14: Señales de eje sano con mucho ruido procesadas mediante el filtro Smooth, en el dominio temporal y en el dominio de la frecuencia.....	57
Figura 4.15: Comparación de la FFT de la señal de eje sano con mucho ruido sin filtrar y procesada mediante el filtro Smooth.....	58
Figura 4.16: Señales de eje con defecto con poco ruido procesadas mediante el filtro Smooth, en el dominio temporal y en el dominio de la frecuencia.....	58
Figura 4.17: Señales de eje con defecto con mucho ruido procesadas mediante el filtro Smooth, en el dominio temporal y en el dominio de la frecuencia.....	59
Figura 4.18: Señales de eje sano con poco ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo y en el dominio de la frecuencia.....	60
Figura 4.19: Señales de eje sano con mucho ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo y en el dominio de la frecuencia.....	61
Figura 4.20: Señales de eje con defecto con poco ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo y en el dominio de la frecuencia.....	62
Figura 4.21: Comparación de las FFT de las señales con poco ruido de eje sano y con defecto procesadas mediante el filtro Hilbert.....	63
Figura 4.22: Señales de eje con defecto con mucho ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo y en el dominio de la frecuencia.....	64
Figura 4.23: Señales de eje sano con poco ruido procesadas mediante el filtro Wavelet, en el dominio temporal y en el dominio de la frecuencia.....	66
Figura 4.24: Señales de eje sano con mucho ruido procesadas mediante el filtro Wavelet, en el dominio temporal y en el dominio de la frecuencia.....	67
Figura 4.25: Comparación de la FFT de la señal de eje sano con poco ruido y con mucho ruido filtradas mediante TW.....	68

Figura 4.26: Señales de eje con defecto con poco ruido procesadas mediante el filtro Wavelet, en el dominio temporal y en el dominio de la frecuencia.....	69
Figura 4.27: Señales de eje con defecto con mucho ruido procesadas mediante el filtro Wavelet, en el dominio temporal y en el dominio de la frecuencia.....	70
Figura 5.1: Toolbox de la red neuronal para señales sin filtrar (caso 1).....	75
Figura 5.2: Gráfica del Rendimiento durante el entrenamiento para señales sin filtrar (caso 1).....	76
Figura 5.3: Gráficos del Estado de Entrenamiento para señales sin filtrar (caso 1)....	77
Figura 5.4: Matrices de confusión de la red neuronal para señales sin filtrar (caso 1).	78
Figura 5.5: Toolbox de la red neuronal para señales procesadas con el filtro Smooth (caso 1).....	80
Figura 5.6: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Smooth (caso 1).....	81
Figura 5.7: Matrices de confusión de la red neuronal para señales procesadas mediante el filtro Smooth (caso 1).....	82
Figura 5.8: Toolbox de la red neuronal para señales procesadas con el filtro Hilbert (caso 1).....	84
Figura 5.9: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Hilbert (caso 1).....	85
Figura 5.10: Matrices de confusión de la red neuronal para señales procesadas mediante el filtro Hilbert (caso 1).....	86
Figura 5.11: Toolbox de la red neuronal para señales procesadas con el filtro Wavelet (caso 1).....	88
Figura 5.12: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Wavelet (caso 1).....	89
Figura 5.13: Gráfico del Estado de Entrenamiento de la red neuronal para señales procesadas con el filtro Wavelet (caso 1).....	90
Figura 5.14: Matrices de confusión de la red neuronal para señales procesadas mediante el filtro Smooth (caso 1).....	90
Figura 5.15: Toolbox de la red neuronal para señales sin filtrar (caso 2).....	93
Figura 5.16: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales sin filtrar (caso 2).....	94
Figura 5.17: Gráfica de acierto para señales sin filtrar (caso 2).....	95

Figura 5.18: Toolbox de la red neuronal para señales procesadas con el filtro Smooth (caso 2).....	96
Figura 5.19: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Smooth (caso 2).....	97
Figura 5.20: Gráfica de acierto para señales procesadas con el filtro Smooth (caso 2).....	98
Figura 5.21: Toolbox de la red neuronal para señales procesadas con el filtro Hilbert (caso 2).....	99
Figura 5.22: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Hilbert (caso 2).....	100
Figura 5.23: Gráfica de acierto para señales procesadas con el filtro Hilbert (caso 2).....	101
Figura 5.24: Toolbox de la red neuronal para señales procesadas con el filtro Wavelet (caso 2).....	102
Figura 5.25: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Wavelet (caso 2).....	103
Figura 5.26: Gráfica de acierto para señales procesadas con el filtro Wavelet (caso 2).....	104

1. INTRODUCCIÓN

El mantenimiento de los componentes mecánicos de cualquier sistema adquiere una gran importancia en el mundo de la ingeniería. De éste depende el correcto funcionamiento de las máquinas a lo largo del tiempo, para lo cual es necesario conocer el estado de los distintos componentes. La investigación llevada a cabo en este proyecto se centra en el campo del mantenimiento industrial.

La meta más importante de cualquier programa de mantenimiento es la eliminación de los desarreglos de la maquinaria [1]. Las averías graves pueden llegar a causar daños serios periféricos a la máquina, lo cual incrementa los costes de reparación.

El segundo propósito del mantenimiento es anticipar y planificar con precisión los requerimientos de los sistemas. De este modo, se pueden reducir los inventarios de repuestos y programar los paros del sistema en los momentos menos perjudiciales.

La tercera meta es incrementar la disponibilidad para la producción del sistema, buscando reducir la cantidad de paros no programados durante su funcionamiento. Así, se pretende mantener la capacidad operacional del sistema reduciendo el tiempo de inactividad de los componentes críticos. En el caso ideal se deben conocer y documentar las condiciones de operación de todos los componentes.

Tipos de mantenimiento

Hay cuatro tipos de mantenimiento: correctivo, preventivo, predictivo y proactivo. El primer tipo de mantenimiento que apareció es el **correctivo**, que consistía en reparar las averías que se producían en la máquina. La máquina se hacía trabajar hasta que una falla interrumpía el servicio. Evidentemente, este sistema tenía unos costes muy altos, además de parones impredecibles en funcionamiento de las máquinas. Además, antes de romper la máquina no funciona bien durante bastante tiempo.

Para paliar estos problemas se empezó a llevar a cabo el mantenimiento **preventivo**, en el que las máquinas son desarmadas y reacondicionadas periódicamente siguiendo programas planificados previamente. De esta manera, al reacondicionar las máquinas antes de que termine su duración esperada se evita que presenten fallas en servicio.

1. Introducción

El mantenimiento preventivo consiguió grandes avances a nivel económico, pero en algunos casos se cambiaban piezas que estaban en perfecto estado o se producían fallas antes de lo previsto. Así surgió la necesidad de crear el mantenimiento **predictivo**, en el que se evalúa el estado de la máquina para sólo cambiar los componentes que empiezan a funcionar mal. Para ello es necesario llevar un seguimiento del funcionamiento de la máquina, pero se evitan los costes de cambiar componentes que no necesitan ser renovados.

La última innovación es el mantenimiento **proactivo**, que sigue la técnica llamada “Análisis de Causas Fundamentales de Faltas”, la cual consiste en buscar la causa fundamental de cada falta en una máquina con el fin de corregirla. De esta forma se evita que la falta vuelva a aparecer [1].

Los mantenimientos predictivo y proactivo no sustituyen a los mantenimientos más básicos (correctivo y preventivo) sino que tratan de reducir al mínimo los costes derivados de ellos. Los programas de mantenimiento modernos incluyen el mantenimiento preventivo, predictivo y proactivo; pero inevitablemente a veces hay que recurrir al correctivo por fallas inesperadas. El objetivo es reducir estas fallas al mínimo y en algunos casos no se puede permitir que ocurra ninguna falla en servicio, ya que podría poner en peligro la seguridad de las personas.

Las técnicas del mantenimiento predictivo también se utilizan para la diagnosis de equipos nuevos o recién reparados, permitiendo determinar si se encuentran en buen estado.

Existe una gran variedad de técnicas de diagnosis aplicables, como el análisis de la temperatura, el análisis de la emisión acústica, el análisis de la composición de los lubricantes, el análisis vibratorio, etc. La técnica más utilizada en los programas de mantenimiento es la medición de vibraciones debido a que la mayoría de los equipos de la industria son mecánicos y rotativos [2].

Estudios efectuados determinan que cerca del 90% de las averías en máquinas rotativas vienen precedidas de aumentos en sus niveles de vibración. Por tanto, existe una relación directa entre las vibraciones medidas y los defectos de las máquinas rotativas [3]. Cada sistema posee una curva característica vibratoria en condiciones de

funcionamiento normales. Cuando comienza a desarrollarse un defecto el comportamiento dinámico se altera y consecuentemente varía su curva vibratoria. Además, cada tipo de defecto también posee su propia curva vibratoria característica, con lo que a menudo se pueden identificar a partir de ésta.

Esto nos lleva a las mayores ventajas del mantenimiento predictivo: que detectan un gran número de averías en su fase inicial, a menudo no es necesario parar la máquina y además permiten determinar la causa de la avería.

1.1 Antecedentes y motivación

Este proyecto se encuentra englobado en un amplio estudio para la diagnosis de ejes de ferrocarril a partir de sus vibraciones, llevado a cabo por el grupo de investigación Maqlab. El estudio citado se encuentra en el área del análisis predictivo.

La importancia de este proyecto y el estudio que lo engloba radica en la implementación de estos sistemas de diagnosis de ejes en los propios ferrocarriles, con el fin de poder controlar el estado de los ejes de las ruedas en todo momento sin necesidad de realizar paradas rutinarias de mantenimiento. Actualmente, el sistema que se lleva a cabo para el mantenimiento de los ejes de ferrocarriles consiste en realizar paradas cada cierto tiempo o número de kilómetros para someterlos a pruebas que determinen su estado y determinar si es necesario cambiarlos o pueden seguir trabajando hasta la siguiente revisión.

Por tanto, la intención del estudio es sustituir el mantenimiento preventivo actual por un mantenimiento predictivo más eficiente y menos costoso. Uno de los inconvenientes de este proceso es la importancia de acertar en la diagnosis del eje, ya que no se puede permitir ningún fallo a la hora de detectar un eje con defectos porque su rotura en servicio pondría en peligro la vida de personas. El mayor problema para conseguir los niveles de efectividad deseados es el ruido presente en las señales de vibración.

Este motivo lleva a la búsqueda de algún filtro que permita facilitar el trabajo a los sistemas de clasificación, para ir acercando la efectividad del método de diagnosis al 100%.

1.2 Objetivo

El objetivo del presente proyecto es el análisis de distintos filtros aplicados a señales vibratorias con el fin de determinar cuál es el más adecuado para detectar fallos en ejes de ferrocarril.

El filtrado es una operación de procesamiento de la señal en el dominio del tiempo en la que se da una importancia predeterminada a diferentes regiones de frecuencia. De esta forma se separan las componentes de la señal, concentradas en distintas regiones de frecuencia, y se disminuye la cantidad de ruido mediante la eliminación de componentes no deseadas [4].

Las señales vibratorias son filtradas inicialmente en el dominio del tiempo y transformadas posteriormente al dominio de la frecuencia con el fin de mejorar la identificación de los posibles defectos.

Los diferentes filtros se han aplicado a señales medidas en el taller de la Universidad Carlos III de Madrid con un eje a escala 1/8.

Las señales se someten a 3 filtros diferentes:

- El filtro Smooth, que suaviza la señal disminuyendo los picos para que la señal sea más homogénea [5]
- Filtrado mediante la Transformada de Hilbert.
- Filtrado utilizando la Transformada Wavelet Estacionaria.

El procesamiento para convertir la señal al dominio de la frecuencia se realiza mediante la Transformada Rápida de Fourier.

El problema de los ejes de ferrocarril es que se desconocen las frecuencias críticas de los ejes sanos y de cada tipo de defecto. Por tanto, es necesario crear una red neuronal y entrenarla con señales de ejes con un estado conocido (eje sano o eje con grieta de profundidad 50% del diámetro) y determinado filtrado. A continuación se probará la red con otras señales del mismo eje, estado conocido y el mismo filtrado y

1. Introducción

se comprobará si la red identifica correctamente los ejes en buen estado y los ejes con defecto.

La eficacia del filtrado se determina en función del porcentaje de acierto de la red neuronal. A mayor porcentaje de acierto, mejor es la eliminación de ruido y por tanto se identifican mejor las frecuencias críticas de los ejes sanos o con defecto.

1.3 Estructura del documento

Capítulo 1: Introducción

La introducción enmarca el proyecto que se va a llevar a cabo en el área del mantenimiento, hablando de los tipos de mantenimiento y la importancia de cada uno. La parte más importante de este capítulo es la explicación de los objetivos del proyecto, donde se indica además la finalidad del estudio llevado a cabo y cómo se va a realizar. También se incluye la estructura del documento en este apartado.

Capítulo 2: Marco teórico

El marco teórico presenta los fundamentos teóricos de todos los procesamientos y transformaciones de la señal que se van a llevar a cabo en el proyecto. Además, se explica en qué consisten los sistemas clasificadores inteligentes y las redes neuronales artificiales.

Capítulo 3: Sistema experimental de medida

En este capítulo se describe cómo funciona el sistema experimental de medida de las señales vibratorias y sus componentes.

Capítulo 4: Procesamiento de las señales y análisis preliminares

Este capítulo consiste en seleccionar unas pocas señales para someterlas a los distintos procesamientos y filtros con el fin de ver el efecto que tienen sobre ellas. De estos resultados se sacarán unas conclusiones preliminares.

Capítulo 5: Redes neuronales artificiales

En esta parte del proyecto se crean, entrenan y testean las redes neuronales. De ellas se obtienen las tasas de acierto para cada filtrado de señal y se consiguen los resultados finales.

Capítulo 6: Conclusiones

Se determina cuál es el filtrado más adecuado para el estudio llevado a cabo a la vista de los resultados obtenidos.

Capítulo 7: Referencias

Consta de la bibliografía utilizada en el proyecto.

Capítulo 8: Anexos

Se incluyen unos anexos en los que se indican las líneas de código utilizadas en Matlab para obtener el procesamiento y filtrado de la señal.

2. MARCO TEÓRICO

2.1 Aplicación del procesamiento de la señal

El principal objetivo del estudio de señales es extraer funciones o parámetros característicos del modelo, con el fin de clasificar, monitorizar y diagnosticar distintos tipos de sistemas.

En este caso, el procesamiento de la señal se utiliza para analizar el comportamiento de sistemas, como máquinas, estructuras o componentes, con el fin de diagnosticar su estado. Tal como se puede observar en la figura 2.1, la excitación de sistemas mediante distintas señales nos facilita datos de su comportamiento.

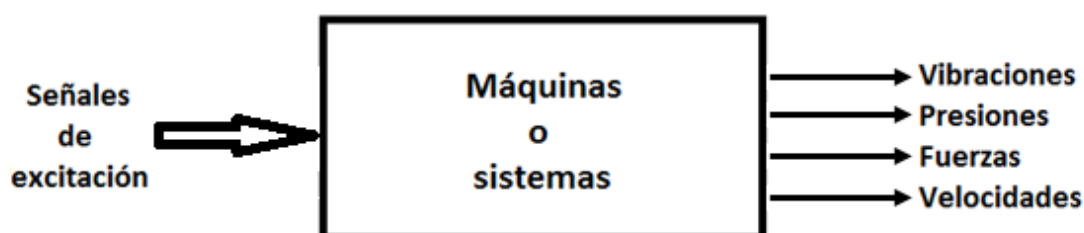


Figura 2.1: Excitación de sistemas para obtener sus datos.

En los apartados siguientes se explicará el estudio de las diferentes técnicas de procesamiento y filtrado de la señal, con el fin de obtener información representativa a partir de las señales de vibración de los rodamientos. Estas técnicas se clasifican dependiendo de si trabajan en el dominio de la frecuencia, el dominio temporal o en ambos dominios.

2.2 El dominio de la frecuencia

El análisis del dominio de la frecuencia es predominante en muchos ámbitos de aplicación, lo cual se debe principalmente a las siguientes razones [4]:

- El significado físico suele ser más fácil de obtener en el dominio de la frecuencia, al contrario del dominio original de tiempo, que describe señales y sistemas. Un claro ejemplo es la existencia de vibraciones periódicas en máquinas rotativas.
- Es más sencillo obtener los patrones de señales para el diagnóstico. Pequeños cambios que son detectados en el dominio de la frecuencia apenas afectan al dominio del tiempo.
- Los sistemas mecánicos se suelen modelar mediante sistemas lineales, quedando así definidos por ecuaciones diferenciales lineales. Al aplicar la Transformada de Fourier, éstas se convierten en ecuaciones algebraicas.
- La disponibilidad de la Transformada Rápida de Fourier, la cual es un algoritmo esencial para el procesamiento de señales.

El análisis de la señal en el dominio de la frecuencia consiste en su descomposición en series de Fourier.

2.3 La Transformada de Fourier

La Transformada de Fourier [6] es la herramienta que nos posibilita describir una función periódica (transitorio), con el objetivo de descomponer dicha función como suma de armónicas. De este modo, una señal temporal $x(t)$ de período T puede expresarse como se observa en la ecuación 2.1:

$$x(t) = a_0 + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi kt}{T} + b_k \sin \frac{2\pi kt}{T} \right) \quad (2.1)$$

2. Marco teórico

Donde los coeficientes vienen dados por las ecuaciones 2.2, 2.3 y 2.4:

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt \quad (2.2)$$

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos \frac{2\pi kt}{T} dt \quad (2.3)$$

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin \frac{2\pi kt}{T} dt \quad (2.4)$$

La Transformada de Fourier también se puede expresar en su forma integral [4], como se muestra en las ecuaciones 2.5 y 2.6:

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp(j2\pi kf) df \quad (2.5)$$

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp(-j2\pi kft) dt \quad (2.6)$$

Expresado simbólicamente $x(t) \leftrightarrow X(f)$, siendo $x(t)$ la representación de la señal en el dominio del tiempo y $X(f)$ en el dominio de la frecuencia. De este modo, se puede convertir una señal que se encuentra en el dominio del tiempo en una que se encuentra en el dominio de la frecuencia y viceversa.

Un ejemplo para entender mejor el uso de la Transformada de Fourier se observa en la figura 2.2, en la que se puede observar una comparación de las dos representaciones mencionadas anteriormente. En la imagen se muestran dos señales sinusoidales de distinta frecuencia. La tercera señal es la suma de las dos primeras y al aplicar la transformada de Fourier se observan las frecuencias de ambos armónicos, mientras que en la representación en el dominio del tiempo sería más complicado reconocerlas.

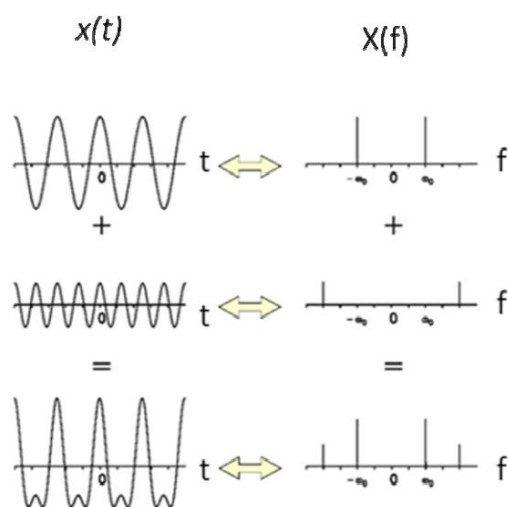


Figura 2.2: Análisis tiempo-frecuencia.

Esto sucede con una señal que se obtiene como suma de solo dos armónicos y normalmente tratamos con señales complejas compuestas por un gran número de ellos. En estos casos, el análisis en el dominio de la frecuencia nos permite distinguir fácilmente la frecuencia de los principales armónicos, cosa que sería muy complicada en el dominio del tiempo.

Por último, la Transformada de Fourier se puede escribir de forma compleja, de forma que se expresa de una forma más compacta, como se puede observar en la ecuación 2.7:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt \quad (2.7)$$

2.4 Transformada Discreta de Fourier (DFT)

La Transformada de Fourier se utiliza cuando la señal es continua en el tiempo, es decir, cuando va de $-\infty$ a ∞ . Por tanto, para poder procesar una señal es necesario muestrearla tomando sólo una parte de ella. Se toman N elementos, siendo N el número de muestras.

Así apareció la Transformada Discreta de Fourier (DFT), mostrada en la ecuación 2.8:

$$X_k = \frac{1}{N} \sum_{r=0}^{N-1} x_r e^{-i2\pi kr/N}; k = 0, 1, 2, \dots, (N-1) \quad (2.8)$$

2.5 Transformada Rápida de Fourier (FFT)

El nombre genérico de la Transformada Rápida de Fourier [7] cubre una familia de algoritmos desarrollados para la computación rápida y eficiente de la DFT.

El proceso matemático para calcular la DFT es muy complejo y puede resultar excesivamente largo, ya que para obtener N componentes en frecuencias de N muestras hay que realizar N^2 multiplicaciones complejas. Por ello fue necesaria la creación de la FFT, que obtiene los resultados de la DFT pero en $N \cdot \log_2 N$ multiplicaciones complejas, reduciendo de forma considerable el número de cálculos. Esto se debe a que subdivide la serie completa en sucesiones más cortas para finalmente combinarlas entre sí, obteniendo la serie original. Por tanto, no calcula la DFT de la serie original sino de otras más sencillas.

En la mayoría de los casos, la FFT funciona mejor si el número de muestras (N) tiene un valor igual a una potencia de 2: $N = 16, 32 \dots 1024, 2048$, etc.

2.6 Transformada Corta de Fourier (STFT)

La Transformada de Fourier presenta problemas a la hora de ser utilizada para analizar señales no estacionarias. Dado que la mayoría de las señales que se generan en la industria tienen carácter no estacionario, esto se convierte en un gran inconveniente. El problema radica en que el espectro de frecuencias variará para distintos instantes de tiempo. Por tanto, dentro de una misma señal nos encontraríamos con varios espectros de frecuencia.

2. Marco teórico

Por tanto, la representación en el espectro de la frecuencia no nos daría información fidedigna de la señal. Por este motivo se crea la idea de la Transformada Corta de Fourier. Esta nueva transformada sigue un esquema parecido al de los diferenciales: considera la señal no estacionaria como la suma de un conjunto de señales adyacentes, que son consideradas como cuasi estacionarias.

Es necesario considerar un intervalo de tiempo, dividiendo la señal en un número finito de señales en ese periodo y aplicado la transformada de Fourier a cada uno de ellos, como muestra la ecuación 2.9:

$$S_x(t, f) = \int_{-\infty}^{\infty} x(u)h(u - t) \exp(-j2\pi ft) du \quad (2.9)$$

En la que $x(u)$ es la señal a analizar y $h(u - t)$ la ventana de tiempo que se traslada a lo largo de la señal. Esta expresión es la que se conoce como transformada de Fourier de tiempo reducido.

El problema de la transformada de Fourier de tiempo reducido es que tiene una precisión dependiente del intervalo de tiempo. Si éste tuviera una longitud infinita podríamos obtener una representación de la frecuencia perfecta pero no tendríamos información temporal alguna. A medida que el intervalo se reduce se aumenta la información en el dominio del tiempo, a la vez que disminuye la información sobre la frecuencia.

2.7 Transformada Wavelet

El análisis Wavelet permite analizar tanto señales estacionarias como no estacionarias, por lo que se convierte en el siguiente paso en la evolución del procesamiento de señales. El análisis que proporciona presenta una resolución múltiple (en tiempo y frecuencia), a diferentes escalas y resoluciones, lo que le convierte en una herramienta especialmente útil para la diagnosis de defectos.

La Transformada Wavelet traslada una señal del dominio espacial al dominio en escala mediante el uso de un grupo infinito de funciones base con energía finita,

2. Marco teórico

llamadas Wavelets. El término "Wavelet" significa "onda pequeña", ya que es un diminutivo de la palabra "wave", que significa onda en inglés. Este nombre hace referencia a la naturaleza oscilatoria de la señal y su longitud finita.

Las Wavelets son familias de funciones que se encuentran definidas en el espacio y se emplean como funciones de análisis, examinando la señal de interés en el plano tiempo-frecuencia para obtener sus características periódicas y no periódicas. Una familia de Wavelets viene definida por la ecuación 2.10:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbb{R} \quad a > 0 \quad (2.10)$$

Cada uno de los componentes de la familia se genera a partir de la Wavelet Madre $\psi(t)$, a través de la variable de escala a , la cual indica las dilataciones y contracciones de la señal, y la traslación b , que se encarga de mover la señal en el tiempo. Las principales familias de Wavelets son Daubechies, Biortogonal, Coiflet, Symlet, Morlet, Sombrero Mexicano y Meyer [8]. Cada una de ellas tiene una ecuación característica que las representa y las da forma, como se puede observar en la figura 2.3 [9].

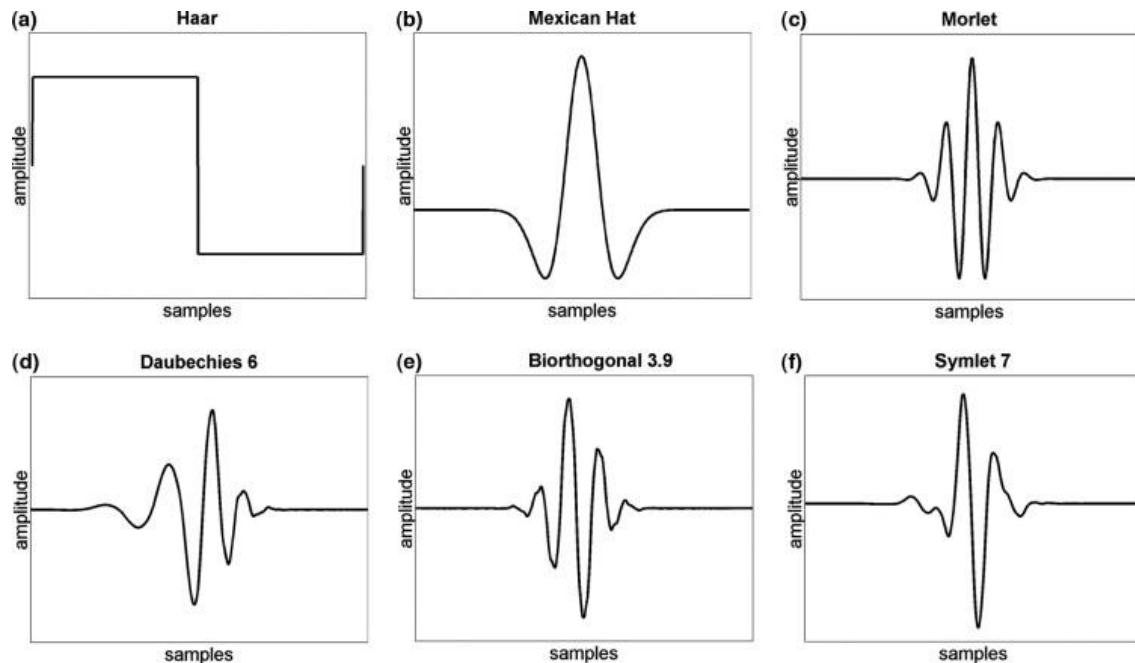


Figura 2.3: Principales familias de Wavelets.

2.8 Transformada Discreta Wavelet (DWT)

Las Transformadas Discretas Wavelet [10] permiten analizar señales tanto en el dominio del tiempo como en el dominio de la frecuencia. Esto se debe a que la DWT divide la señal en bandas de frecuencias formadas por coeficientes wavelet. Los coeficientes wavelet se corresponden con los datos de la señal temporal que mejor representan esa banda de frecuencias. Cuando las amplitudes de los coeficientes son grandes indican que su banda de frecuencia es importante dentro de la señal y que las amplitudes en el dominio del tiempo a las que pertenece también lo son.

Las bandas de frecuencia se obtienen haciendo pasar la señal temporal por una serie de filtros de paso bajo y paso alto con el fin de descomponer la señal. De esta forma se obtiene una serie de muestras que corresponden a una banda de frecuencias en el dominio del tiempo. En el primer nivel de la transformada wavelet, la señal pasa por un filtro paso bajo, obteniendo los llamados coeficientes de aproximación, y por un filtro paso alto, consiguiendo los coeficientes de detalle. En los siguientes niveles, las señales que pasan los filtros paso bajo y paso alto son tan solo las correspondientes a los coeficientes de aproximación. En cada nivel de descomposición, el número de muestras que componen cada banda de frecuencias es la mitad que en el anterior.

En la figura 2.4 se muestra un esquema del funcionamiento de la DWT y su descomposición en los coeficientes de aproximación y coeficientes de detalle. Los coeficientes de aproximación se indican mediante la letra "A", mientras que los coeficientes de detalle están señalados con la letra "C". El número que acompaña a estas letras indica el nivel de descomposición.

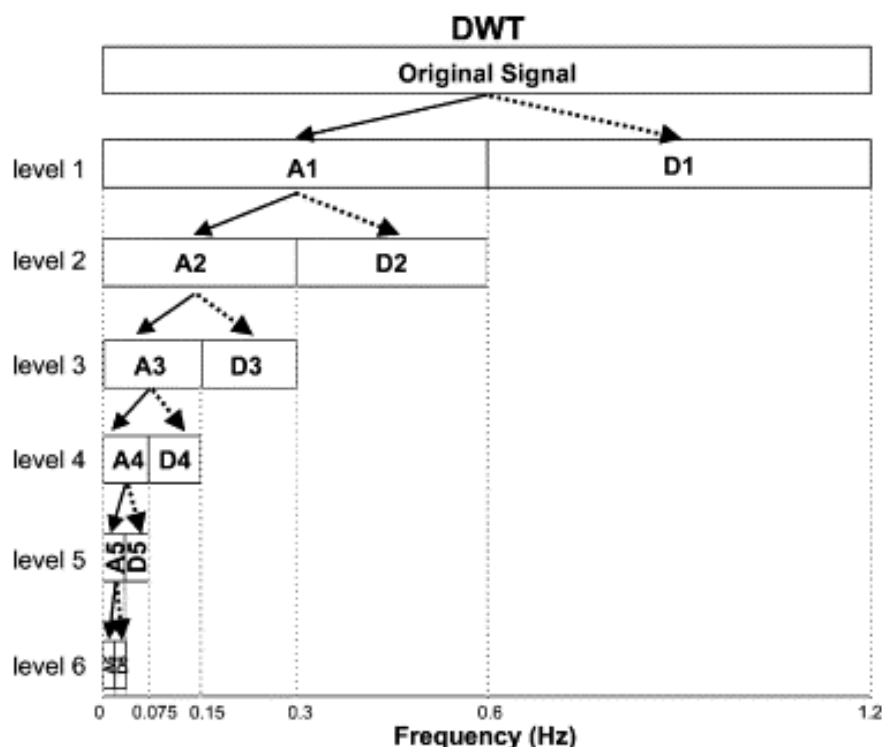


Figura 2.4: Esquema del funcionamiento de las DWT.

2.9 Transformada Wavelet Packets (WPT)

Las Transformadas Wavelet Packets [11] son un tipo de DWT especial, que se diferencian en que no sólo se dividen las señales de los coeficientes de aproximación en cada nivel de descomposición, sino que también se dividen las señales de los coeficientes de detalle.

En la figura 2.5 se muestra un esquema de la descomposición mediante wavelet packets. Se parte de la señal original y se va dividiendo en cada nivel. En cada señal, el primer subíndice indica el nivel de descomposición y el segundo, la posición dentro de cada nivel. La cantidad de componentes asociada a un nivel de descomposición viene dada por la expresión 2^j , siendo j el número del nivel. Así, se puede comprobar que en el nivel 3 se tienen $2^3 = 8$ componentes.

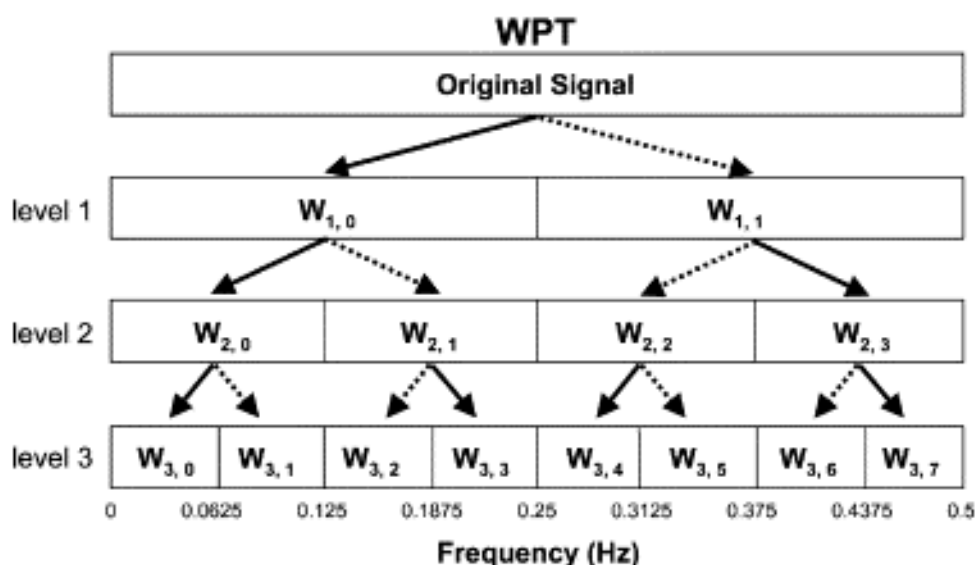


Figura 2.5: Esquema del funcionamiento de las WPT.

Además, en la figura 5 se observa que en la señal original fue tomada a una frecuencia de muestreo de 0,5 Hz. El primer nivel de descomposición está formado por dos señales: una de ellas comprendida entre 0 y 0,25 Hz y la otra entre 0,25 y 0,5 Hz. A su vez, el segundo nivel está formado por 4 señales de rangos de frecuencia: 0 - 0,125 Hz, 0,125 - 0,25 Hz, 0,25 - 0,375 Hz y 0,375 - 0,5 Hz.

2.10 Transformada de Hilbert

La transformada de Hilbert [6] es un método adaptativo, capaz de tratar señales que contienen datos no estacionarios y no lineales. La mayoría de las señales reales presentan estas características, lo que confiere a la Transformada de Hilbert una gran utilidad.

Además, algunas señales ocultan información debido al ruido y las interferencias que perturban la señal. La Transformada de Hilbert demodula la señal para tratar de eliminar el ruido presente en el banco de ensayos y así obtener espectros de frecuencia más claros al realizar la FFT.

2. Marco teórico

La expresión analítica de la Transformada de Hilbert [12] viene dada por la ecuación 2.11:

$$y(t) = H[x(t)] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(u)}{t-u} du \quad (2.11)$$

La señal analítica $z(t)$ viene definida por la ecuación 2.12:

$$z(t) = x(t) + jy(t) = a(t)e^{j\theta(t)} \quad (2.12)$$

El término $a(t)$ corresponde a la señal envolvente de $x(t)$, y su expresión viene definida por la ecuación 2.13:

$$a(t) = \sqrt{x^2(t) + y^2(t)} \quad (2.13)$$

El término $\theta(t)$ es la señal de fase instantánea de $x(t)$ y se define con la ecuación 2.14:

$$\theta(t) = \arctg\left(\frac{y(t)}{x(t)}\right) = 2\pi f_0 t \quad (2.14)$$

Donde la frecuencia instantánea f_0 viene dada por la ecuación 2.15:

$$f_0 = \left(\frac{1}{2\pi}\right) \frac{d\theta(t)}{dt} \quad (2.15)$$

2. Marco teórico

En la figura 2.6 se puede observar un ejemplo de aplicación de la transformada de Hilbert a una señal $x(t)$ cuadrada de 2 segundos de periodo.

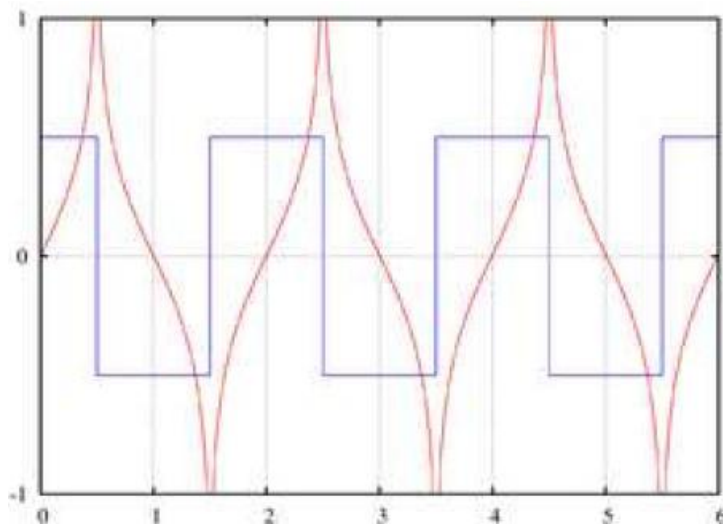


Figura 2.6: Transformada de Hilbert de una onda cuadrada.

La transformada de Hilbert suele utilizarse a modo de filtro para obtener la envolvente de la respuesta del sistema. Posteriormente, se aplica a la señal la Transformada Corta de Fourier.

2.11 Sistemas clasificadores inteligentes

El reconocimiento de patrones (PR) es una actividad inconsciente que los seres humanos realizamos con total naturalidad. Procesamos inmediatamente la información que recibimos a través de nuestros sentidos, siendo capaces de identificar la fuente de información casi al instante. De hecho, la mayoría de las tareas que realizamos diariamente están basadas en nuestra capacidad de realizar distintas actividades de reconocimiento de patrones. Por ejemplo, al leer reconocemos los caracteres, formando palabras y a continuación conceptos y nociones de la señal visual que recibe nuestro cerebro. El PR es el acto de ejecutar una acción basada en las características de la información percibida.

Por este motivo, se intentan diseñar Sistemas Clasificadores Inteligentes capaces de reconocer patrones. La Clasificación inteligente ha sido un factor clave en el desarrollo de los métodos de diagnóstico de defectos [13], pero no ha madurado completamente

2. Marco teórico

debido al continuo aporte de ideas de disciplinas como la neurobiología, física o psicología.

Un Sistema Clasificador Inteligente se encarga de asignar una clase a todos los objetos que se deseen analizar, de acuerdo a sus diferencias o similitudes, utilizando los patrones de un conjunto de muestras. Este proceso se divide en dos etapas: la primera consiste en el aprendizaje de propiedades de un grupo de muestras que caracterizan a una clase (patrones), mientras que en la segunda se determina si un objeto distinto es un posible miembro de tal clase, observando si éste presenta propiedades comunes a las del grupo de muestras.

El principal objetivo en este campo es el desarrollo de nuevos algoritmos y metodologías que sean capaces de implementar muchas de las tareas de PR que efectúan las personas, con el fin de suplantarlos en tareas de PR que sean repetitivas o mecánicas, cada vez con más precisión y sin la necesidad de un experto.

2.12 Redes neuronales artificiales (ANN)

Las redes neuronales artificiales [14] son un tipo de Sistema Clasificador muy extendido. Ofrecen respuesta a los problemas que requieren una cantidad de recursos prácticamente imposible para obtener una solución exacta. Sus aplicaciones industriales más comunes son: identificación y modelado, optimización, clasificación, control de procesos y reconocimiento de patrones[15].

Las ANN aparecieron al intentar simular los sistemas nerviosos biológicos. La primera red neuronal artificial fue propuesta por McCulloch y Pitts en 1943 [16], al intentar emular la actividad nerviosa. Esta ANN consistía en un modelo binario formado por un conjunto de unidades llamadas "neuronas" conectadas entre sí, en el cual cada neurona tiene un umbral preestablecido. Estas conexiones presentan un gran parecido a las dendritas y los axones de los sistemas nerviosos biológicos. Las similitudes y correspondencias entre redes neuronales artificiales y biológicas se pueden observar en la figura 2.7.

2. Marco teórico

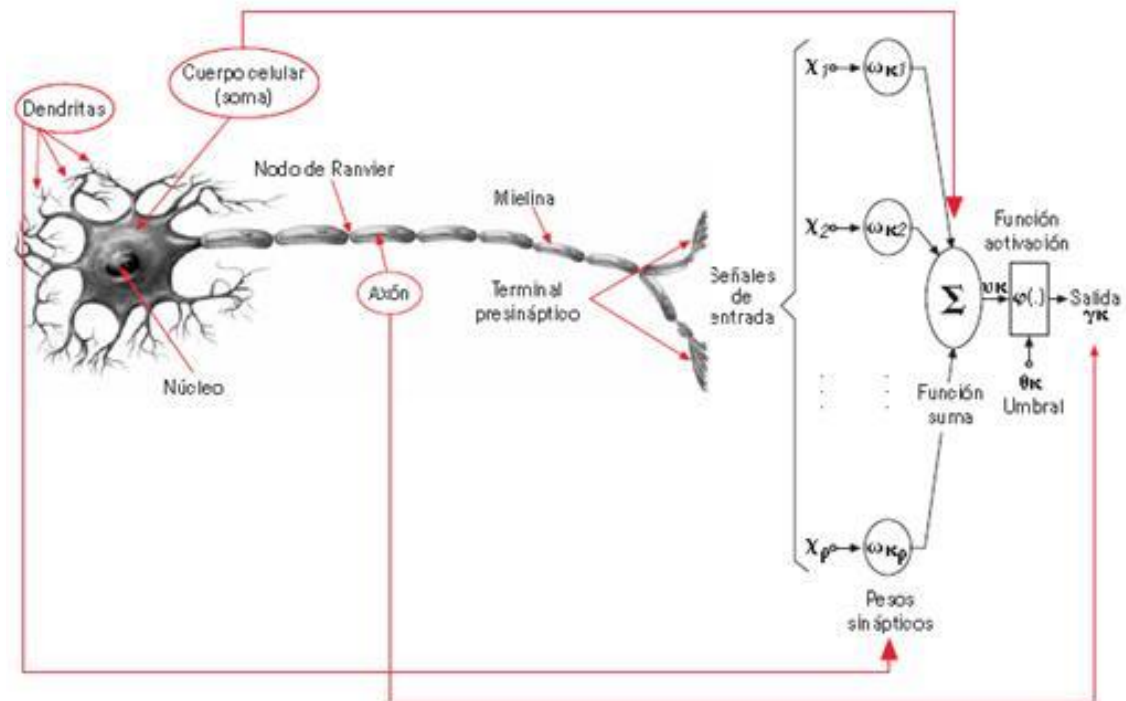


Figura 2.7: Comparación entre red neuronal biológica y artificial.

Las partes fundamentales de una neurona artificial básica [17] se pueden observar en la figura 2.7, y son:

- **Entrada:** Es el vector que excita a la neurona. Se compone de los datos más representativos (patrones) del fenómeno al que se desea aplicar una ANN.
- **Pesos:** Son valores que se asocian a cada una de las conexiones entre neuronas de una red. La variación de éstos determina el aprendizaje de una red.
- **Función de activación:** Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total. Es la característica que mejor define el comportamiento de las neuronas. Existen diferentes funciones de activación aplicadas a distintos modelos de ANN.
- **Función de red o de propagación:** Calcula el valor base o entrada total a la unidad y equivale a la combinación de las señales que excitan o inhiben a las neuronas biológicas.
- **Salida:** Es el valor de salida de la neurona en función de la activación de la misma. En el caso de la utilización de una ANN para el proceso de diagnóstico de fallos, la salida equivale al diagnóstico de la red.

2. Marco teórico

Las ANN básicas tienen tres capas de neuronas como mínimo: una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa de la ANN contiene un grupo de unidades de procesamiento, llamadas neuronas o nodos, que envían su propia salida a todas las neuronas de la capa siguiente de manera unidireccional. En la figura 2.8 se puede observar una red neuronal básica con " n " neuronas en su capa de entrada, una capa oculta con " m " neuronas y una sola neurona en su capa de salida.

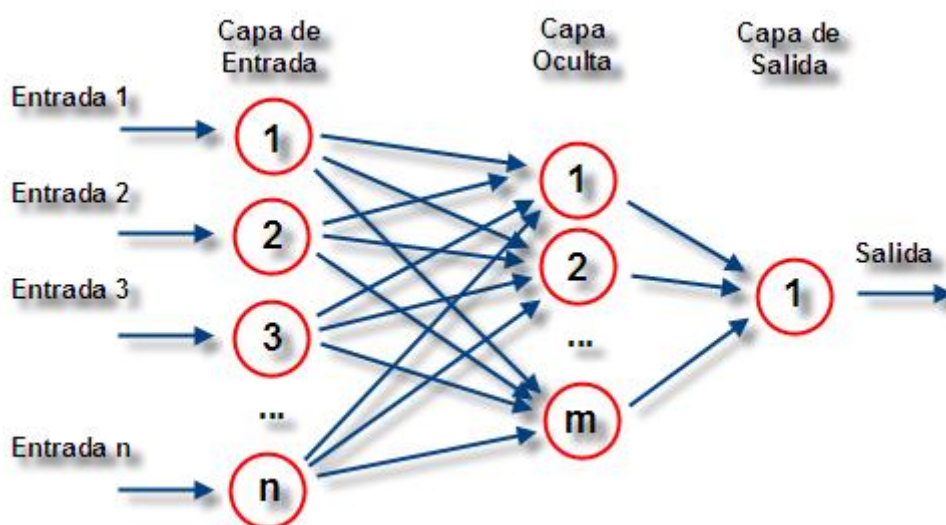


Figura 2.8: Red neuronal básica de tres capas.

Las señales externas son recibidas a través de las neuronas de la capa de entrada, que sirven de puente entre el mundo exterior y las unidades de proceso internas, las neuronas de la capa oculta. La capa exterior, a su vez, sirve como una interfaz entre las unidades internas de la red y el mundo exterior.

El funcionamiento de una ANN convencional se divide en dos fases: la fase de aprendizaje o entrenamiento y la fase de prueba. Una fracción de la información de entrada se utiliza para entrenar la red mientras que la otra parte se utiliza para la fase de prueba.

Las ANN son sistemas de aprendizaje basados en ejemplos. En la fase de aprendizaje, los valores internos de la red van adaptándose progresivamente hasta obtener el comportamiento deseado. Los ejemplos se introducen paulatinamente, modificando el valor de los pesos en todas las conexiones con un algoritmo

2. Marco teórico

determinado, hasta obtener un error que se encuentre dentro de los márgenes de diseño de la red. Este proceso es efectuado de manera cíclica por la red, que va comparando la salida real con la salida esperada. Por tanto, la capacidad de una red para resolver un problema está ligada fundamentalmente al tipo de ejemplos disponibles en el proceso de aprendizaje. El objetivo de esta fase es minimizar el error para las neuronas de salida de la red.

El objetivo principal de la fase de prueba es verificar el comportamiento de datos desconocidos para la red. En esta fase no se alteran los valores de los pesos o las conexiones internas de la red, sino que se mide el éxito real del modelo creado en la fase de aprendizaje.

3. SISTEMA EXPERIMENTAL DE MEDIDA

3.1 Introducción

El presente capítulo describe todo el proceso y la maquinaria necesaria para la toma de señales de vibración del eje sometido a ensayo. Las medidas han sido tomadas en el taller mecánico de la Universidad Carlos III de Madrid, por un grupo de estudiantes entre los que se encuentra el autor del presente proyecto. Las medidas han sido compartidas en varios proyectos, con el fin de aumentar el número de señales para cada estudio.

Para obtener las señales necesarias para todos los proyectos de diagnosis de ejes que estaban en curso durante el período de toma de medidas, se ensaya el eje a tres velocidades de rotación distintas: 20, 40 y 60 Hz (1200, 2400 y 3600 rpm). Además, para poder comparar el eje sano con el mismo eje defectuoso, se realizan primero las medidas con el eje en perfecto estado, después se realiza un corte en el punto medio del eje que simula un defecto, volviendo a tomar las señales a las distintas velocidades de rotación. Este proceso se repite varias veces, alcanzando cuatro estados distintos de defecto:

- Eje sano.
- Eje con defecto al 33% del radio.
- Eje con defecto al 66% del radio.
- Eje con defecto al 100% del radio.

Para el estudio realizado en el presente proyecto se han utilizado solamente las señales de eje sano y eje con defecto al 100% del radio, a 20 Hz de velocidad.

3.2 Componentes del equipo de ensayo

El equipo de ensayo consta principalmente de los siguientes componentes:

- Banco de ensayos.
- Motor eléctrico.
- Rodamientos.
- Eje.
- Panel de control.
- Tacómetro.
- Acelerómetro.
- Amplificador.
- Tarjeta de adquisición de datos.
- Ordenador personal.

A continuación se detallan los componentes mencionados.

BANCO DE ENSAYOS

El banco de ensayos utilizado para obtener las señales vibratorias sometidas a estudio es un "Rotor Kit MFS" [18], que se puede observar en la figura 3.1. Esta Máquina de Simulación de Fallos fue diseñada por la empresa Spectra Quest con el fin de simular o estudiar comportamientos de elementos rotatorios reales o a escala para detectar los fallos de los mismos.

Esta máquina permite realizar experimentos dinámicos, estudios de vibraciones o fallos en maquinaria mediante experimentos controlados, pero necesita otros componentes adicionales específicos para cada tipo de estudio determinado.

3. Sistema experimental de medida

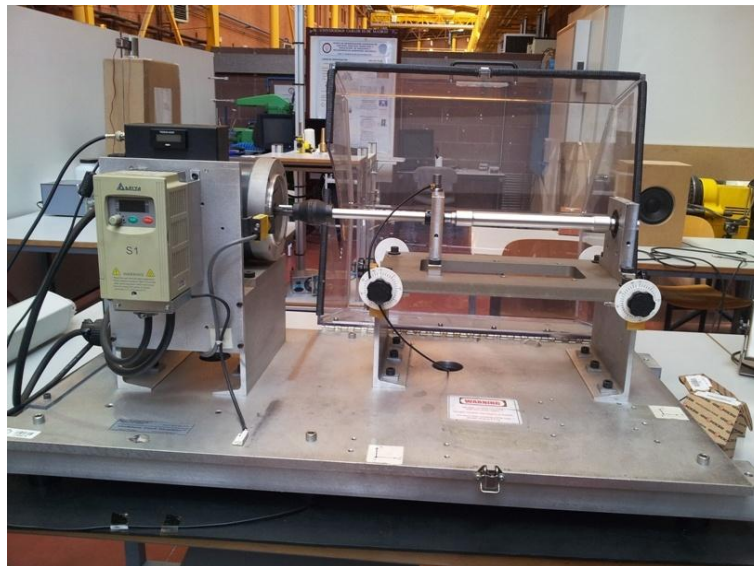


Figura 3.1: Banco de ensayos "Rotor Kit MFS".

MOTOR ELÉCTRICO

Se trata de un motor eléctrico de la marca Marathon Electric (figura 3.2), que trabaja con corriente alterna, dispone de tres fases y es capaz de funcionar a una potencia de 1/2 CV o 1/3 CV. Es el elemento que hace girar al eje sometido a estudio, al que está conectado mediante un acoplamiento.



Figura 3.2: Motor eléctrico Marathon Electric.

RODAMIENTOS

Los rodamientos utilizados para sujetar el eje sometido a estudio son el modelo ER10K (figura 3.3) de la marca MB Manufacturing. El catálogo de dicha compañía especifica todas las características principales de los rodamientos y en la tabla 3.1 se realiza un resumen de las más representativas.

Tabla 3.1: Características principales de los rodamientos ER10K.

Diámetro interior [mm]	15,875
Diámetro exterior [mm]	46,990
Diámetro medio [mm]	33,490
Número de bolas	8
Diámetro de las bolas [mm]	7,940
Ángulo de contacto [°]	0



Figura 3.3: Rodamiento ER10K.

3. Sistema experimental de medida

EJE

Se trata de un eje de ferrocarril a escala 1/8 (figura 3.4), de aluminio. Los defectos se han simulado realizando un mecanizado en la zona media del mismo.



Figura 3.4: Eje de ferrocarril a escala

PANEL DE CONTROL

El panel de control (figura 3.5) se utiliza para controlar la puesta en marcha o parada del motor y su velocidad de rotación. Es de la marca Delta y dispone de una pequeña pantalla que muestra en todo momento la velocidad actual en hertzios.



Figura 3.5: Panel de control Delta.

TACÓMETRO

El tacómetro (figura 3.6) se encarga de medir la velocidad de rotación del eje del motor, en rpm, para asegurar que no hay ningún fallo en el motor o el panel de control. La velocidad del eje siempre es ligeramente menor a la especificada en el panel de control, debido a las pérdidas por rozamiento.



Figura 3.6: Tacómetro digital.

ACELERÓMETRO

El acelerómetro es un instrumento destinado a medir aceleraciones, fundamental para determinar las vibraciones. Consta de un imán para poder fijarlo a elementos metálicos y se colocará sobre uno de los tornillos de los soportes del banco de ensayo. Se trata de un acelerómetro piezoeléctrico, cuyo funcionamiento se basa en que, al comprimirse un retículo cristalino piezoeléctrico, se produce una carga eléctrica proporcional a la fuerza aplicada.

El acelerómetro utilizado es el modelo 4383 de la marca Brüel & Kjaer, cuya sensibilidad es de $3,172\text{pC/ms}^{-2}$, y puede observarse en la figura 3.7.



Figura 3.7: Acelerómetro 4383.

3. Sistema experimental de medida

AMPLIFICADOR

El amplificador está directamente conectado al acelerómetro y su función es amplificar la señal que le llega de éste. Este instrumento es necesario porque la corriente creada por el acelerómetro tiene una intensidad tan baja que no es posible someterla a estudio.

El amplificador utilizado en los ensayos del presente proyecto es el modelo 2693 de la marca Brüel & Kjær (figura 3.8) y el factor de amplificación (ganancia) empleado es de 10.



Figura 3.8: Amplificador 2693.

TARJETA DE ADQUISICIÓN DE DATOS

La tarjeta de adquisición de datos se encarga de transformar las señales analógicas proporcionadas por el amplificador en señales digitales que puedan ser utilizadas por el ordenador.

3. Sistema experimental de medida

La tarjeta de adquisición de datos utilizada es el modelo KUSB-3100 de ocho canales, de la marca Keithley, la cual se puede observar en la figura 3.9.



Figura 3.9: Tarjeta de adquisición de datos KUSB-3100.

ORDENADOR PERSONAL

Las señales llegan al ordenador personal (figura 3.10) directamente de la tarjeta de adquisición de datos. El ordenador se programa para recoger la cantidad de datos deseada.



Figura 3.10: Ordenador personal, tarjeta de adquisición de datos y amplificador.

3. Sistema experimental de medida

CADENA DE MEDIDA

En la figura 3.11 se representa un pequeño esquema de la cadena de medida que se utiliza para la toma de señales.

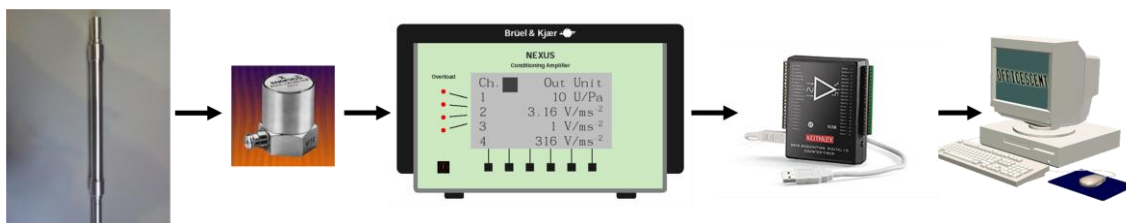


Figura 3.11: Cadena de medida.

3.3 Adquisición de datos

La adquisición de datos se lleva a cabo mediante un módulo de Matlab® llamado "Btool.m" creado por Javier Romero Carrasco [19]. Se trata de una interfaz gráfica para la toma de señales, cuyo aspecto se muestra en la figura 3.12.

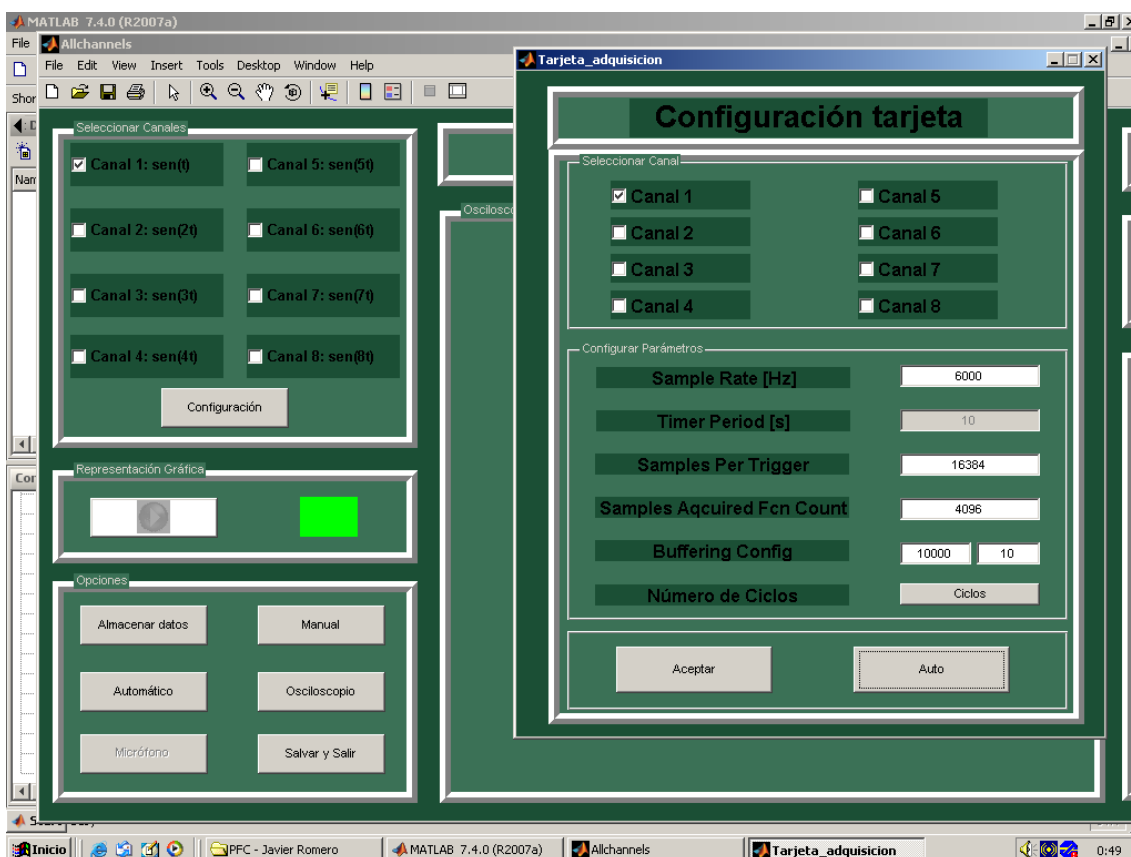


Figura 3.12: Btool.m.

3. Sistema experimental de medida

Se utiliza el modo de adquisición de datos automático, por el canal 1, que es al que está conectada la tarjeta Keithley. La frecuencia de muestreo se establece en 6000 Hz y el número de muestras por señal en 2^{14} . Se configura el programa para que realice 100 mediciones, con un tiempo de un segundo entre cada una y da comienzo la toma de datos.

Pasados los 473 segundos (7 minutos y 53 segundos) que dura la medición de las 100 señales, se muestra en pantalla la representación gráfica de la última señal obtenida, como se observa en la figura 3.13.

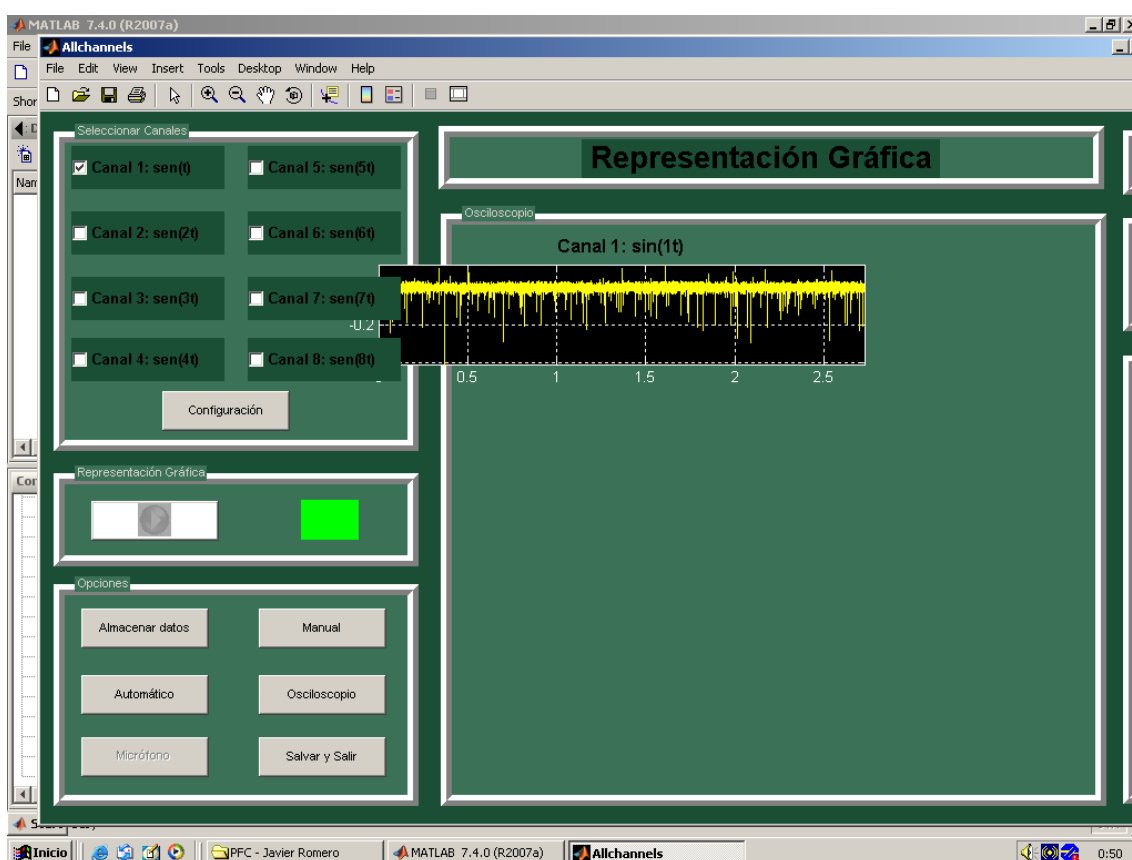


Figura 3.13: Representación gráfica Btool.m

4. PROCESAMIENTO DE LAS SEÑALES Y ANÁLISIS PRELIMINARES

4.1 Introducción

El objetivo de este capítulo es mostrar el efecto de los distintos procesos de filtrado de la señal y comparar los resultados obtenidos.

Todo el procesamiento de las señales se ha realizado mediante el programa Matlab R2010a. Se han creado diversos algoritmos en el lenguaje de Matlab utilizando las funciones preprogramadas para someter a la señal a los procesamientos deseados.

4.2 Selección de señales a estudiar

Como se ha especificado en el apartado 3.3, cada vez que se realizaba una toma de medidas se obtenían 100 señales para cada una de las tres velocidades de rotación del eje. Tras varios meses midiendo, se obtuvieron miles de señales de vibración, con lo que resulta necesario seleccionar solamente algunas señales para realizar un estudio detallado. Por tanto, en este proyecto se van a estudiar sólo las señales a 20 Hz de velocidad de rotación para eje sano y para eje con defecto al 100% del radio.

Además, para el análisis preliminar también fue necesario hacer una selección más reducida de las señales, con el fin de observar en detalle el resultado de cada filtrado. Para ello se seleccionaron 4 carpetas de medidas, de 100 señales cada una. Dos de las carpetas corresponden a eje sano, una con señales muy limpias y la otra con una gran cantidad de ruido. Las otras dos son de eje con defecto al 100% del radio, escogidas también de forma que una presenta una gran cantidad de ruido y la otra el mínimo encontrado.

Finalmente, se seleccionan las tres señales más representativas de cada carpeta. Para ello se realizan tres histogramas de cada carpeta. El primero de ellos representa

4. Procesamiento de las señales y análisis preliminares

el valor medio de cada una de las 100 señales, con el fin de determinar cuál es el más frecuente. Los otros dos representan los valores máximos y mínimos de cada señal. Los histogramas son necesarios para encontrar las tres señales cuyos valores medio, máximo y mínimo se acerquen más a los que se han repetido un número mayor veces.

Otra forma alternativa para tener en cuenta más señales de cada carpeta habría sido hacer la media total de los valores de todas las señales para crear una señal que englobara todas las de la carpeta. Este método es totalmente factible si se hace con las señales en el dominio de la frecuencia pero en el dominio del tiempo el resultado no es correcto, ya que cada señal temporal empieza en una posición distinta del periodo de rotación del eje. Aunque las señales temporales son parecidas en la forma, sus funciones están desplazadas en el eje temporal. Por tanto, se perdería toda la información temporal de las señales, motivo por el cual se descartó esta posibilidad y se decidieron escoger señales sueltas.

La figura 4.1 representa el histograma del valor medio de cada una de las 100 señales con poco ruido de eje sano. Como se puede observar, el más repetido es el $4,5 \cdot 10^{-3}$, valor que tienen como media 8 de las 100 señales estudiadas.

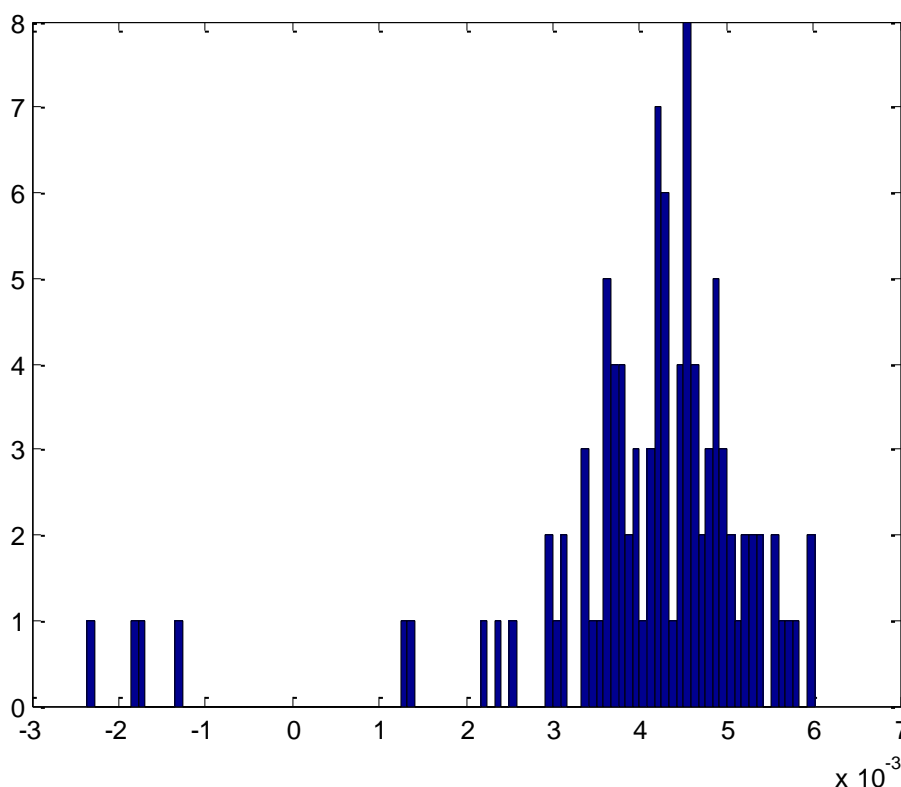


Figura 4.1: Histograma de la media de las 100 señales de eje sano con poco ruido.

4. Procesamiento de las señales y análisis preliminares

En las figuras 4.2 y 4.3 se presentan los histogramas de valores máximos y mínimos respectivamente. Se puede observar que el máximo que más se repite es el valor 0,55, con 7 repeticiones. Para los valores mínimos, el histograma es más confuso, ya que hay varios valores que se repiten el mismo número de veces. En este caso tomamos los valores -0,45 y -0,47, por encontrarse ambos en la zona en la que se concentra la mayor cantidad de repeticiones.

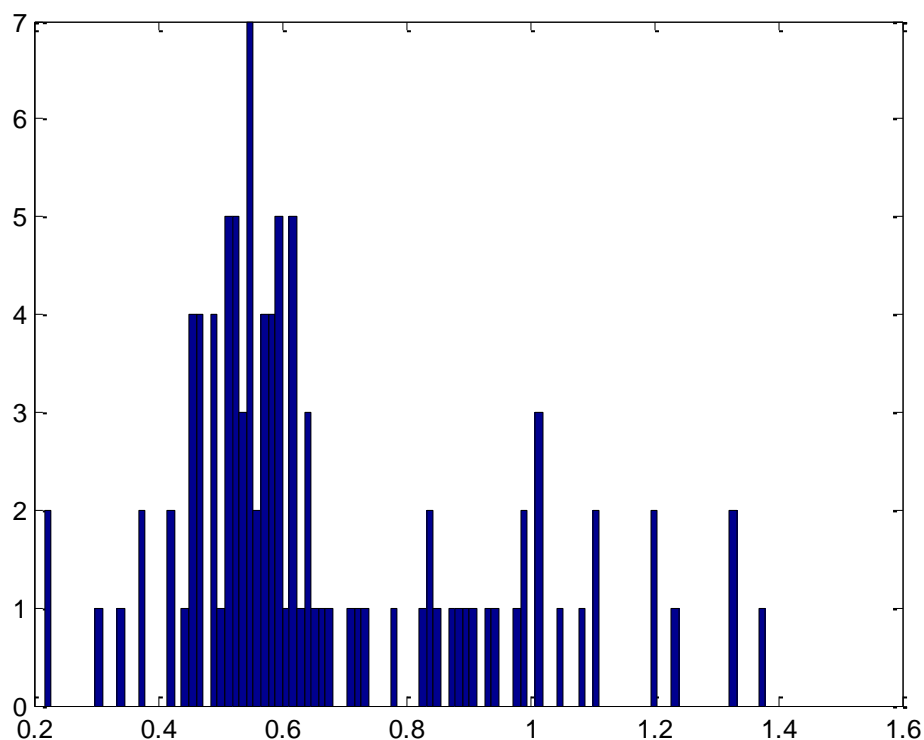


Figura 4.2: Histograma de los máximos de las 100 señales de eje sano con poco ruido.

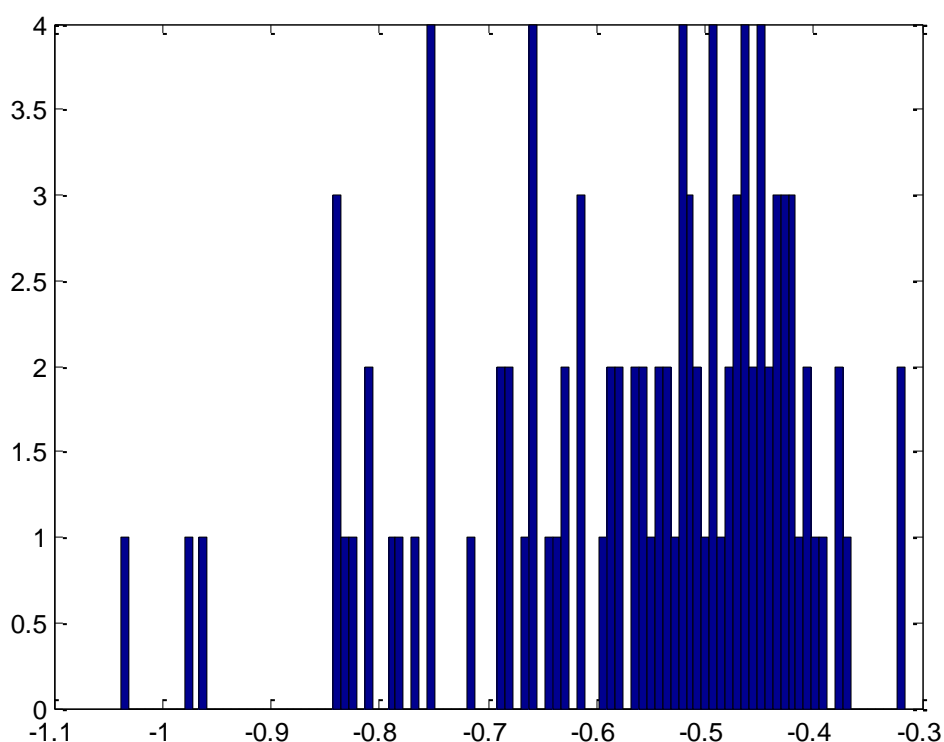


Figura 4.3: Histograma de los mínimos de las 100 señales de eje sano con poco ruido.

Por tanto, se seleccionan las tres señales cuyos valores medio, máximo y mínimo se acerquen más a $4,5 \cdot 10^{-3}$, 0,55 y -0,45/-0,47 respectivamente.

Este mismo proceso se repite para las otras tres carpetas escogidas en este apartado. De nuevo, para seleccionar las señales, se escogen aquellas cuya media, valor máximo y valor mínimo estén entre los más repetidos dentro de su carpeta.

4.3 Señal original

4.3.1 Señal temporal original

En este capítulo se van a presentar las señales temporales originales para observar la diferencia entre las señales del eje sano y el eje con defecto, además de poder apreciar qué señal tiene ruido y cuál está limpia.

4. Procesamiento de las señales y análisis preliminares

Se mostrarán las tres señales representativas seleccionadas, por el método explicado en el capítulo anterior, para cada una de las cuatro carpetas de datos escogidos (eje sano sin ruido, eje sano con ruido, eje con defecto al 100% del radio sin ruido y eje con defecto al 100% del radio con ruido).

En primer lugar, en la figura 4.4 se observan las señales del eje sano con la mínima presencia de ruido conseguida.

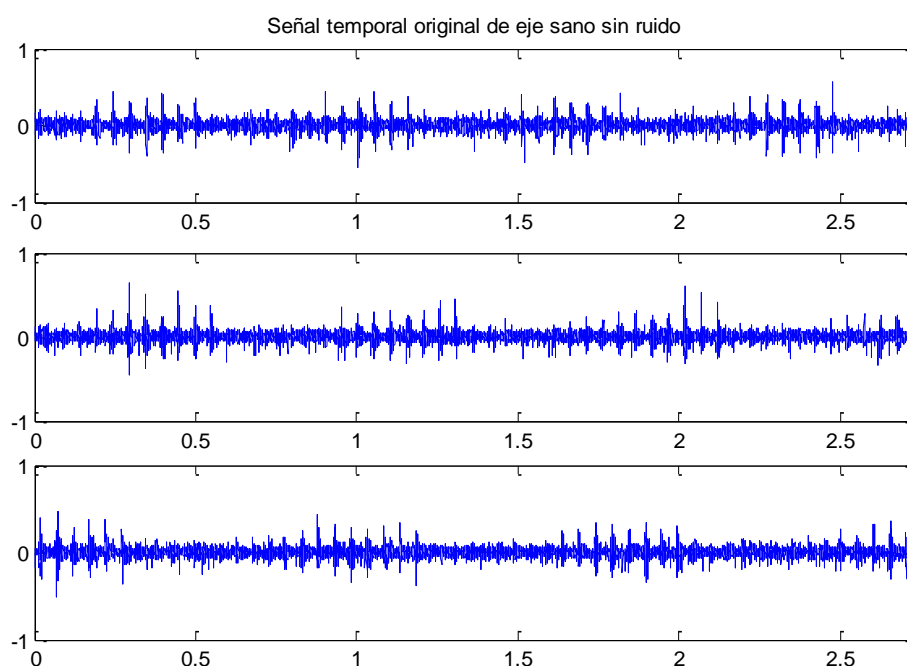


Figura 4.4: Señales temporales de vibración de eje sano con poco ruido.

4. Procesamiento de las señales y análisis preliminares

En segundo lugar, en la figura 4.5 se muestran las señales de eje sano con una gran presencia de ruido.

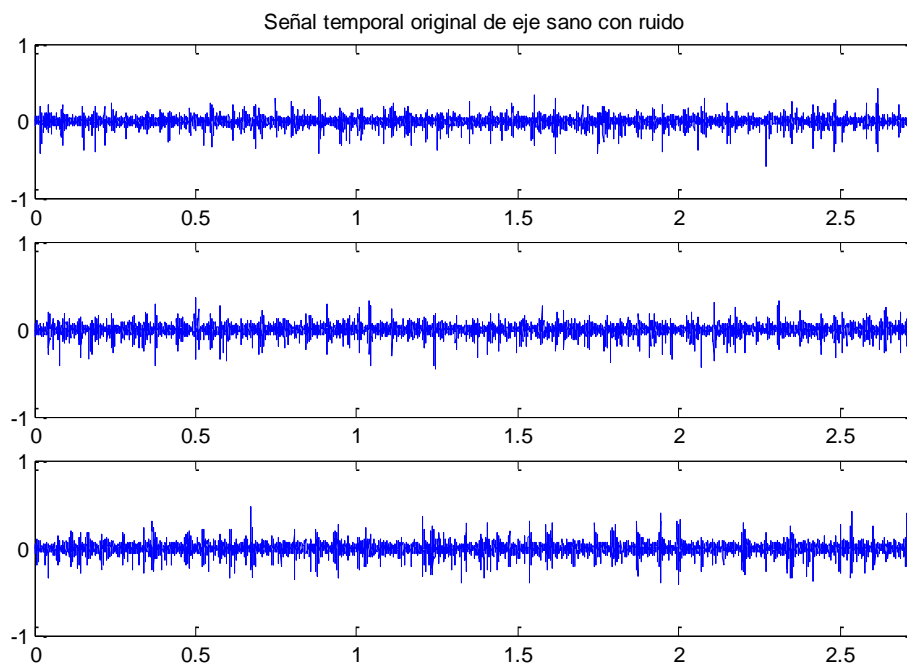


Figura 4.5: Señales temporales de vibración de eje sano con mucho ruido.

En tercer lugar, en la figura 4.6 se presentan las señales de vibración de eje con defecto al 100% del radio sin ruido.

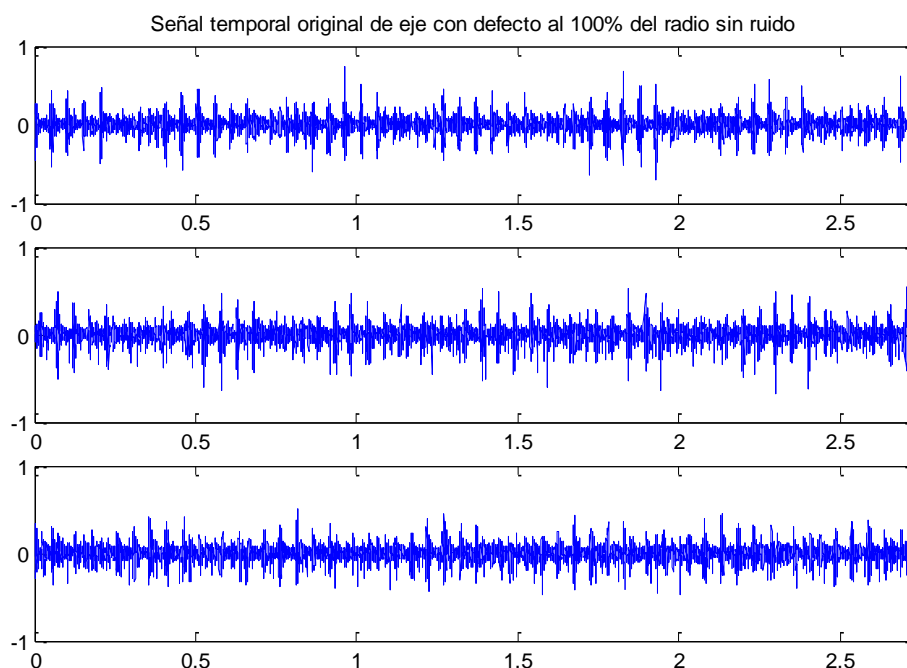


Figura 4.6: Señales temporales de vibración de eje con defecto al 100% del radio con poco ruido.

4. Procesamiento de las señales y análisis preliminares

En cuarto y último lugar, en la figura 4.7 se pueden observar las señales vibratorias de eje con defecto al 100% del radio con ruido.

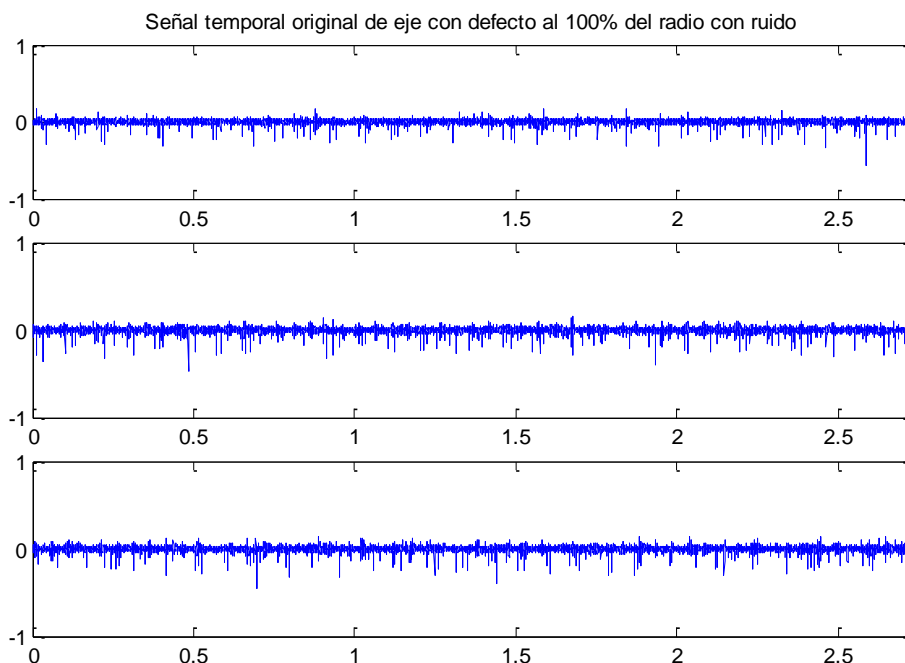


Figura 4.7: Señales temporales de vibración de eje con defecto al 100% del radio con mucho ruido.

Como se observa en las imágenes, es imposible determinar a simple vista qué señales pertenecen a eje con defecto y cuáles a eje sano, y menos aún si tendrán ruido o no tras ser sometidas al procesamiento y transformación al dominio de la frecuencia. Por tanto, en principio las señales temporales no nos aportan información útil para el diagnóstico de ejes.

4.3.2 Señal en el dominio de la frecuencia (FFT)

En el apartado actual se van a mostrar las señales sin filtrar en el dominio de la frecuencia, tras haber sido sometidas a la Transformada Rápida de Fourier. Veremos si estas señales nos aportan alguna información analizándolas a simple vista.

En la figura 4.8 se muestra la FFT de la señal de eje sano sin ruido. Como se puede observar, la señal está muy limpia y se pueden identificar claramente los picos de frecuencia en 622,2 Hz y 327,8 Hz.

4. Procesamiento de las señales y análisis preliminares

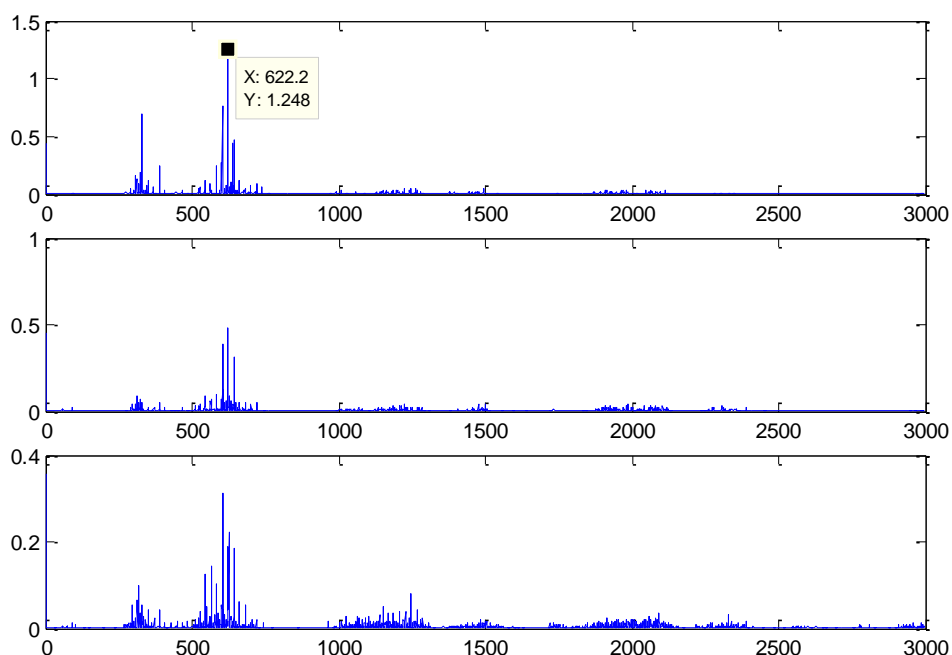


Figura 4.8: FFT de las señales de eje sano con poco ruido.

La FFT de la señal de eje sano con ruido (figura 4.9), en cambio, no tiene picos claramente identificados, sino que presenta subidas en casi todo el rango de frecuencias.

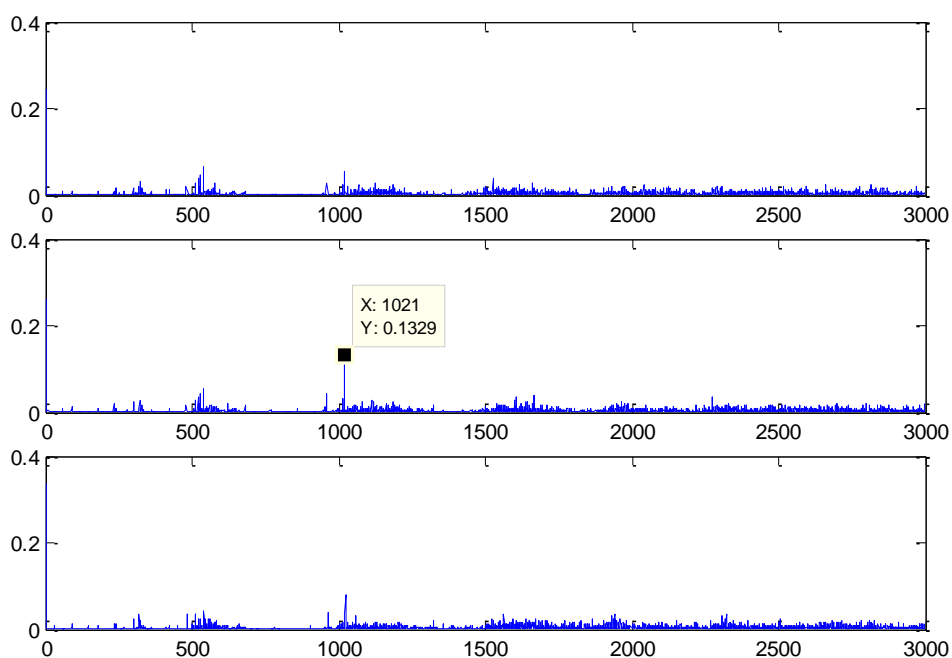


Figura 4.9: FFT de las señales de eje sano con mucho ruido.

4. Procesamiento de las señales y análisis preliminares

Se puede observar que el pico más grande se presenta a una frecuencia de 1021 Hz, lo cual no facilita nada el diagnóstico del eje, ya que no se parece al del mismo eje sin presencia de ruido. A pesar de que las señales son muy distintas, ambas corresponden al mismo eje en perfecto estado y deberían obtener el mismo diagnóstico, lo cual hace necesario un buen filtrado para eliminar las partes de la señal que no interesan para este estudio.

A continuación se procede a analizar la FFT de las señales de eje con defecto, tanto sin ruido (figura 4.10) como con ruido (figura 4.11).

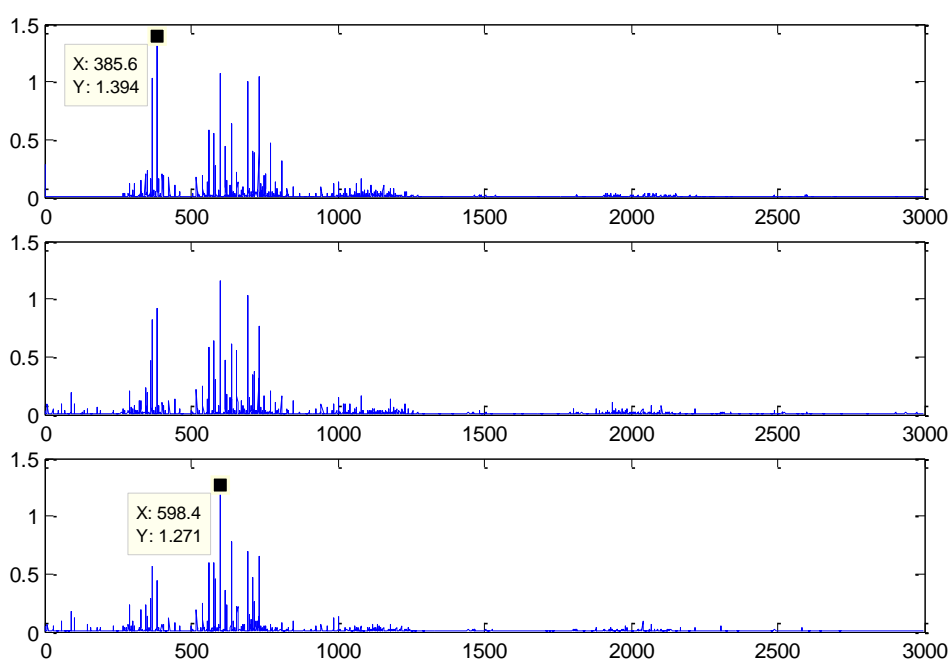


Figura 4.10: FFT de las señales de eje con defecto con poco ruido.

4. Procesamiento de las señales y análisis preliminares

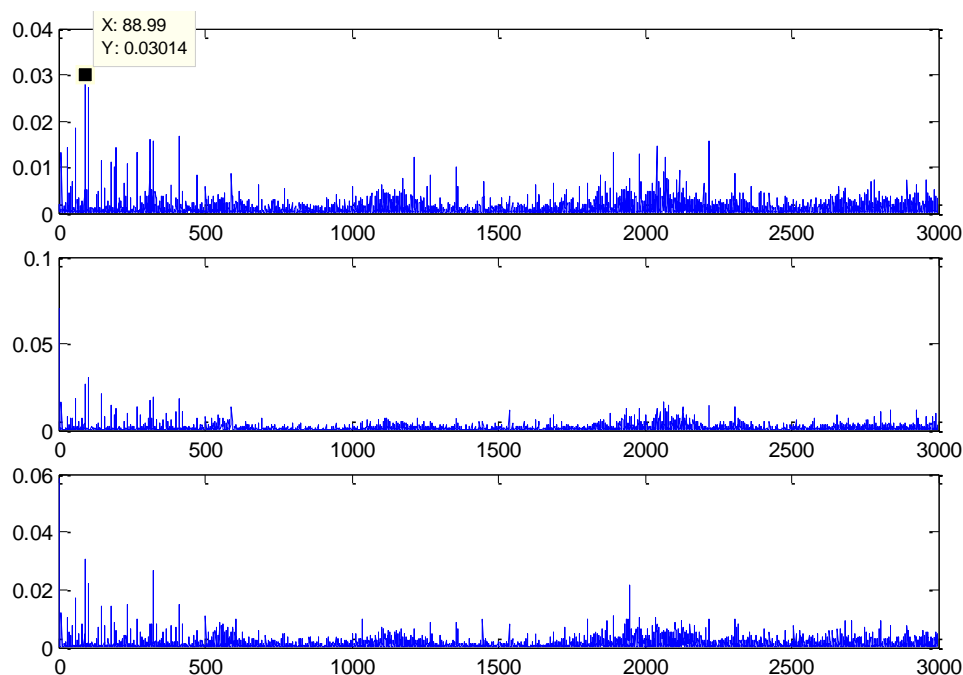


Figura 4.11: FFT de las señales de eje con defecto con mucho ruido.

En las señales sin ruido se pueden observar perfectamente los picos más notables, marcados en la figura 4.10, con unas frecuencias de 385,6 Hz y 598,4 Hz. Éstas deberían ser las frecuencias características del modelo estudiado, que darían la información del estado del eje. Pero, en cambio, en las señales con ruido el pico de frecuencia se sitúa en 88,99 Hz y corresponde al mismo eje con el mismo defecto. Por tanto, el ruido enmascara las frecuencias características que se buscan en la señal y no nos permiten establecer un diagnóstico adecuado.

Por todo esto, en los apartados siguientes se va a proceder a filtrar la señal para comprobar si esto supondrá algún tipo de ayuda en el diagnóstico del eje.

4.4 Filtro Smooth

Al procesar una señal mediante el filtro Smooth, el objetivo es crear una función aproximada que trata de capturar los patrones importantes de la señal, dejando fuera el ruido. El filtro Smooth suaviza la señal utilizando un filtro de media móvil.

Los filtros de media móvil [20] son los más comunes en el procesamiento de señales, más que nada porque es el tipo de filtro digital más fácil de entender y usar. A pesar

4. Procesamiento de las señales y análisis preliminares

de su simplicidad, son óptimos para la tarea de reducir el ruido aleatorio. Ésto los convierte en los mejores filtros para señales en el dominio del tiempo. En cambio, los filtros de media móvil son los peores para señales codificadas en el dominio de la frecuencia, ya que tienen poca habilidad para separar unas bandas de frecuencia de otras.

A continuación se van a mostrar las señales de vibración procesadas mediante el filtro Smooth para ver cómo funciona en la práctica. La figura 4.12 presenta la señal de eje sano con poco ruido, a la izquierda en el dominio del tiempo y a la derecha en el dominio de la frecuencia.

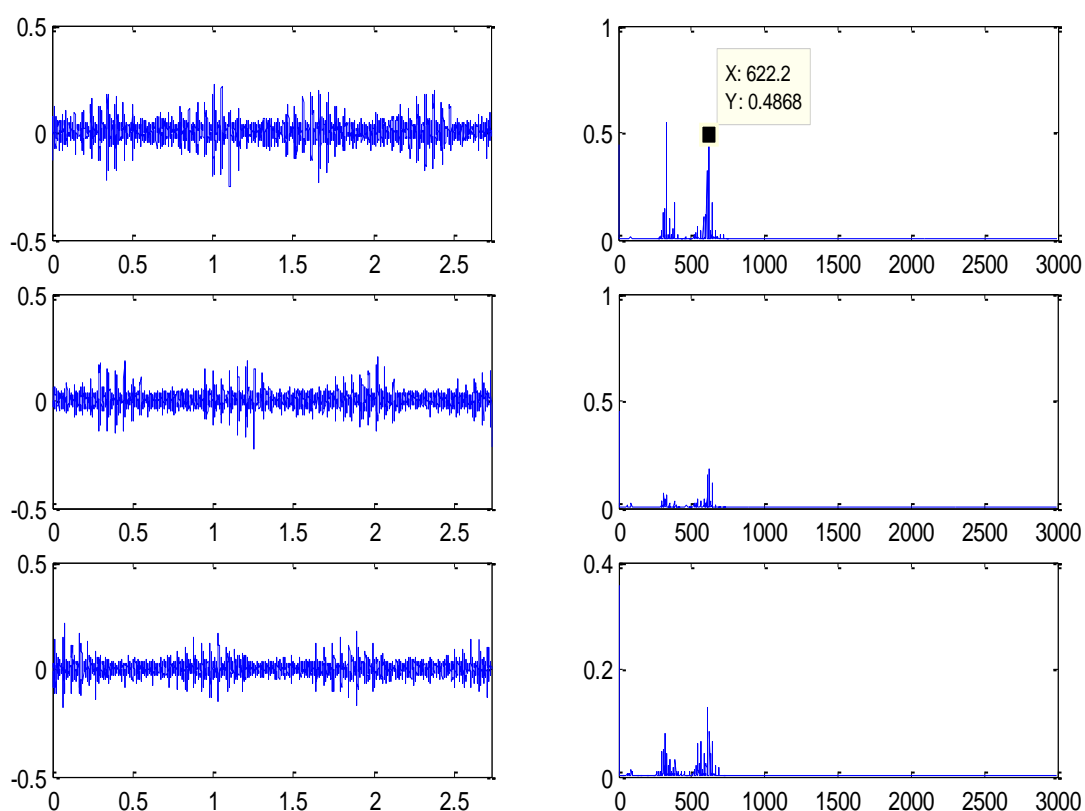


Figura 4.12: Señales de eje sano con poco ruido procesadas mediante el filtro Smooth, en el dominio del tiempo (izquierda) y en el dominio de la frecuencia (derecha).

Se puede observar que las señales temporales tienen patrones de repetición mucho más identificables que antes de ser filtradas, por lo que se distinguen mucho mejor las frecuencias que se repiten en estos gráficos. Se confirma, por tanto, que el dominio del tiempo está mucho más limpio. En el dominio de la frecuencia, el resultado es el mismo que anteriormente, con los picos más pronunciados en los valores 622,2 Hz y

4. Procesamiento de las señales y análisis preliminares

327,8 Hz, pero con una desaparición de los pequeños picos que no eran representativos. En ambos dominios se puede identificar una clara disminución en los valores del eje Y, ya que este filtro se encarga de suavizar la señal.

Para apreciar la mejoría de la señal en el dominio del tiempo se muestran en la figura 4.13 la primera señal sin tratar (arriba) y la primera señal filtrada (abajo).

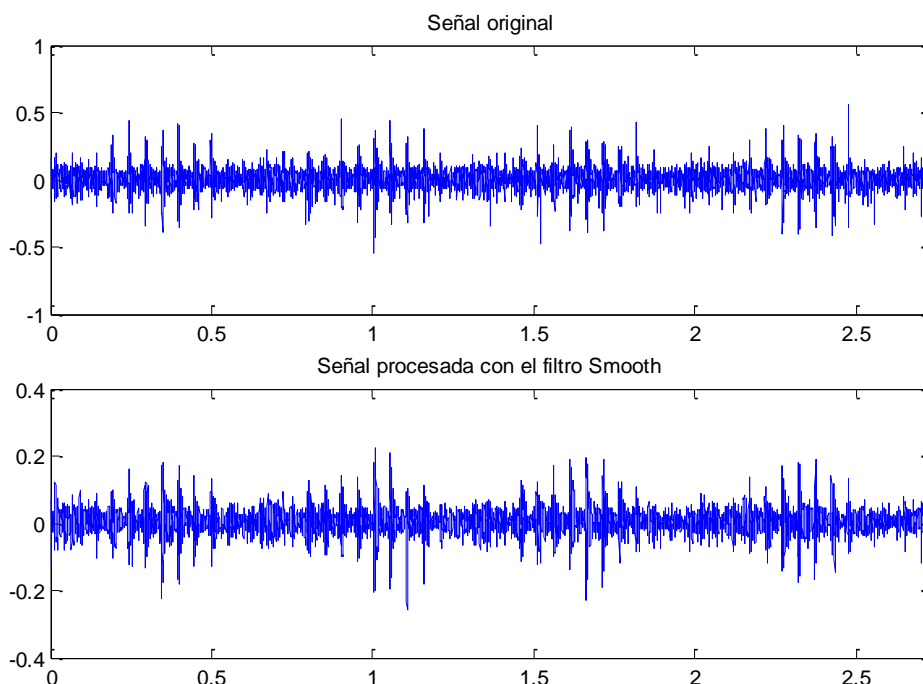


Figura 4.13: Comparación de la señal temporal de vibración de eje sano con poco ruido sin filtrar (arriba) y procesada mediante el filtro Smooth (abajo).

La figura 4.14 representa las señales de eje sano con ruido, a la izquierda en el dominio del tiempo y a la derecha en el de la frecuencia (el esquema de distribución será el mismo en todas las figuras que presenten gráficas distintas a la izquierda y la derecha, a partir de ahora se obviará esta aclaración).

4. Procesamiento de las señales y análisis preliminares

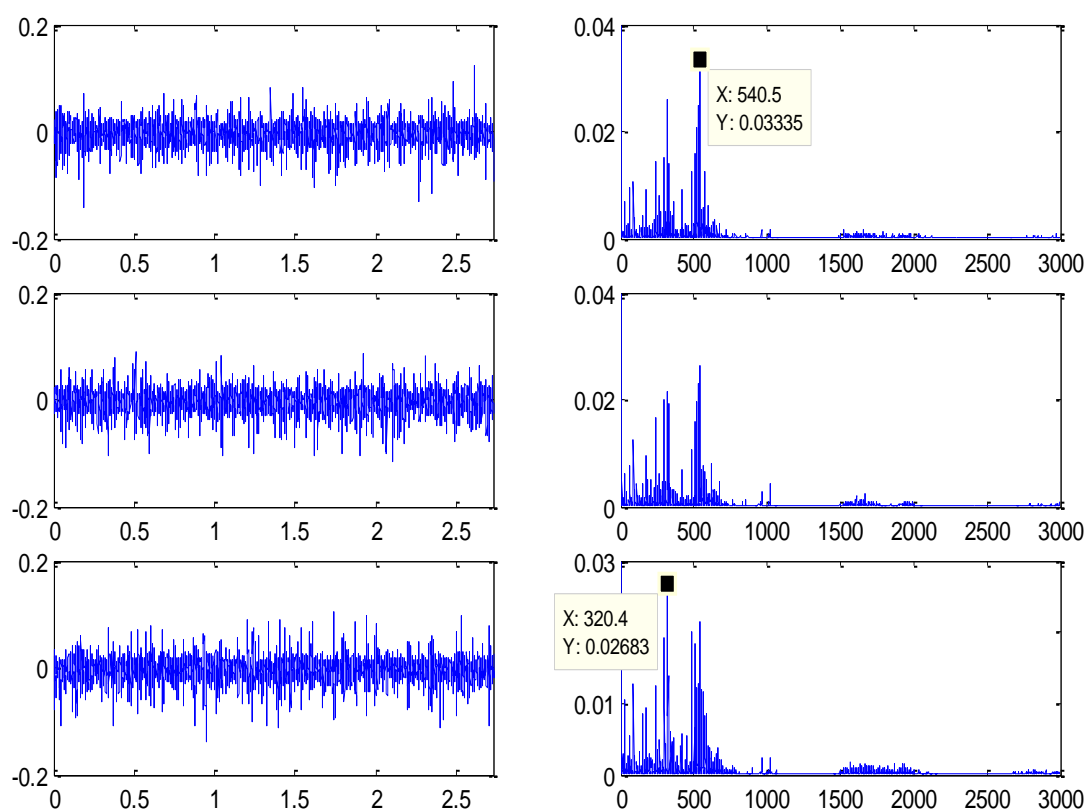


Figura 4.14: Señales de eje sano con mucho ruido procesadas mediante el filtro Smooth, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

En esta ocasión se ha conseguido una reducción de ruido considerable en el dominio de la frecuencia: se han eliminado los picos que presentaba la señal a partir de los 1000 Hz y los nuevos picos máximos (320,4 Hz y 540,5 Hz) se parecen mucho más a los de las señales que se tomaron en ausencia de ruido (327,8 Hz y 622,2 Hz).

A continuación, en la figura 4.15, se muestra una comparación de la señal en el dominio de la frecuencia antes (arriba) y después (abajo) de filtrarla, con el fin de apreciar las diferencias mencionadas anteriormente. Hay una clara disminución del ruido aleatorio.

4. Procesamiento de las señales y análisis preliminares

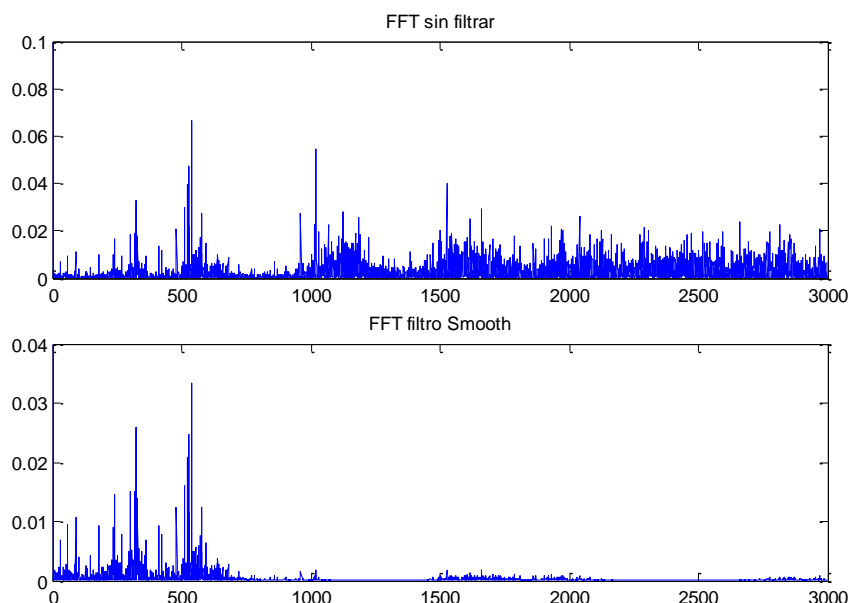


Figura 4.15: Comparación de la FFT de la señal de eje sano con mucho ruido sin filtrar (arriba) y procesada mediante el filtro Smooth (abajo).

El siguiente paso es observar las señales de eje con defecto filtradas. Así, la figura 4.16 representa las señales de eje con defecto con poco ruido procesadas mediante el filtro Smooth.

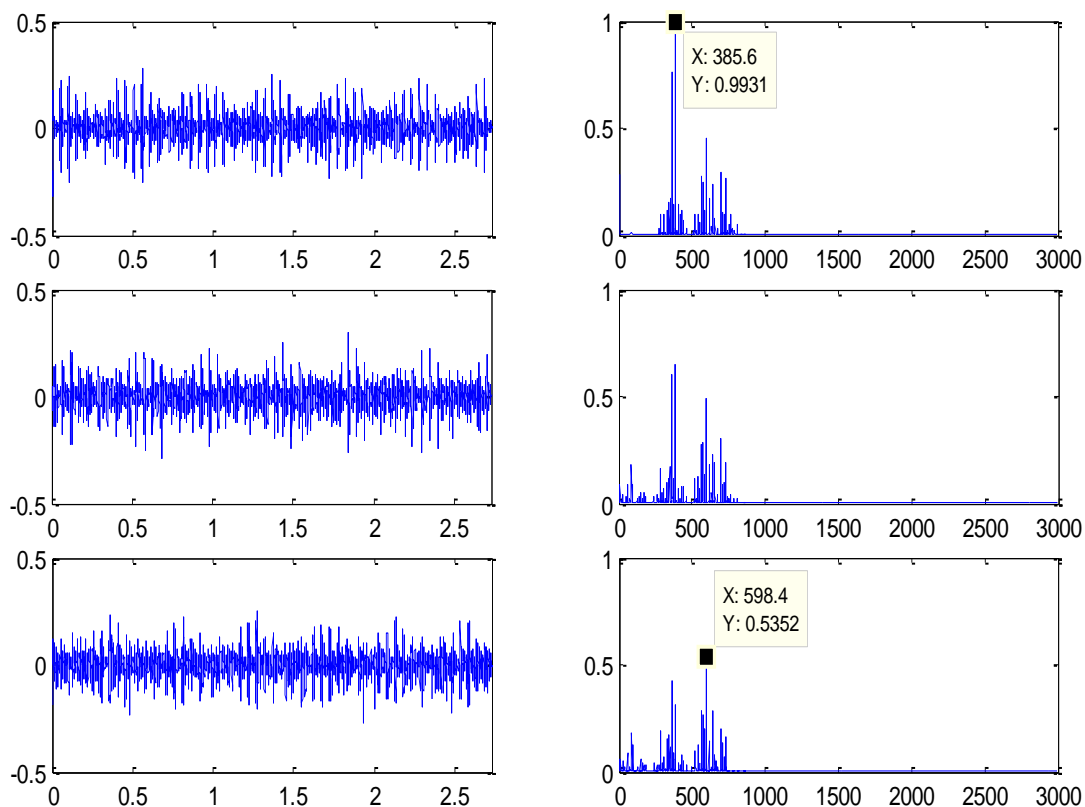


Figura 4.16: Señales de eje con defecto con poco ruido procesadas mediante el filtro Smooth, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

4. Procesamiento de las señales y análisis preliminares

Una vez más, se aprecia una mejoría en la repetición de los patrones de vibración en la señal temporal respecto de la señal sin filtrar. En cambio, en el dominio de la frecuencia no se aprecian grandes cambios y los valores pico de frecuencia son exactamente los mismos que antes de aplicar el filtrado.

Posteriormente, se aplica el filtro a las señales de eje con defecto que tienen una gran presencia de ruido y se muestra el resultado en la figura 4.17.

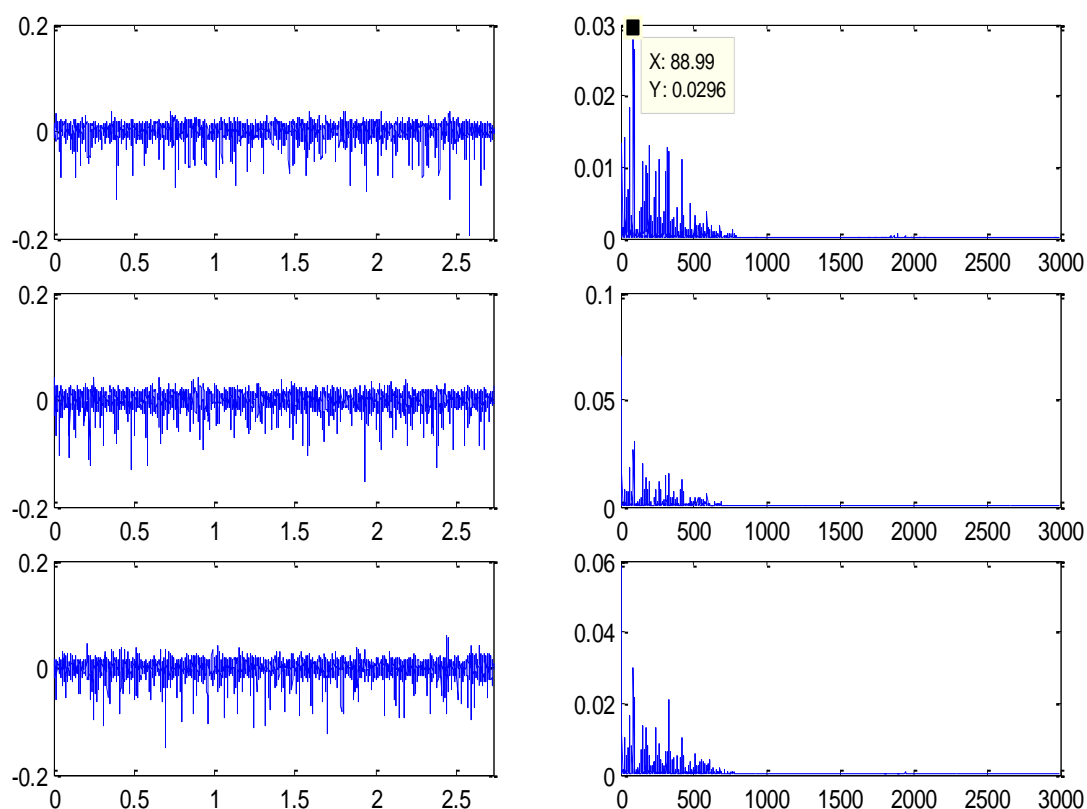


Figura 4.17: Señales de eje con defecto con mucho ruido procesadas mediante el filtro Smooth, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

En este caso, no se observa ninguna mejoría clara ni en el dominio del tiempo ni en el de la frecuencia. El dominio de la frecuencia sigue teniendo los mismos valores pico que antes del filtrado y aunque presenta menos picos aleatorios, los valores siguen sin parecerse a los de la señal de eje con defecto sin ruido. Por tanto, no parece que el filtro Smooth haya tenido un efecto notable en este último caso.

Como conclusiones del análisis preliminar cabría destacar que las señales con poca presencia de ruido mejoran sensiblemente su señal temporal, sin grandes cambios en

4. Procesamiento de las señales y análisis preliminares

su señal en el dominio de la frecuencia. En cambio, las señales con mucho ruido pueden presentar una gran mejoría en el dominio de la frecuencia, acercándose los valores pico a los de las señales limpias; o no presentar mejoría alguna, dependiendo de cada caso.

4.5 Filtrado mediante la Transformada de Hilbert

En este capítulo se va a realizar un análisis preliminar del filtrado mediante la Transformada de Hilbert, con el posterior procesado mediante la FFT. De nuevo, se usarán las señales seleccionadas en el apartado 4.2 para llevar a cabo el estudio.

En primer lugar, en la figura 4.18, se muestran las señales de eje sano con poco ruido, tanto en el dominio del tiempo como en el de la frecuencia.

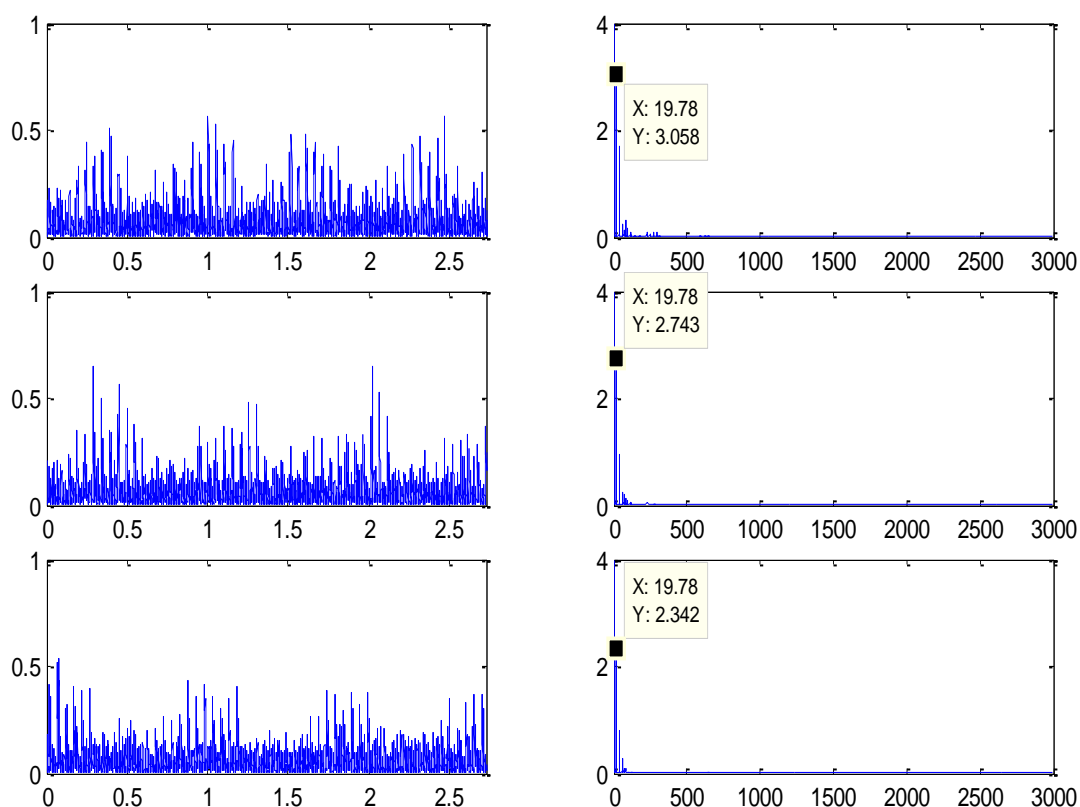


Figura 4.18: Señales de eje sano con poco ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo (izquierda) y en el dominio de la frecuencia (derecha).

4. Procesamiento de las señales y análisis preliminares

El resultado del filtrado mediante la Transformada de Hilbert es bastante notable en el dominio de la frecuencia, obteniéndose una gráfica muy sencilla con picos claramente diferenciados. Las tres señales estudiadas coinciden en que el pico máximo se sitúa en 19,78 Hz y el segundo más grande en 39,55 Hz. En la figura 4.18 es difícil identificar el primer pico, ya que queda tapado por el eje de ordenadas.

A priori, parece que el resultado de este filtrado es muy bueno para señales con poco ruido, pero las señales con gran presencia de ruido son las determinantes para determinar la calidad de un filtro. La figura 4.19 representa las señales de eje sano con mucho ruido aleatorio.

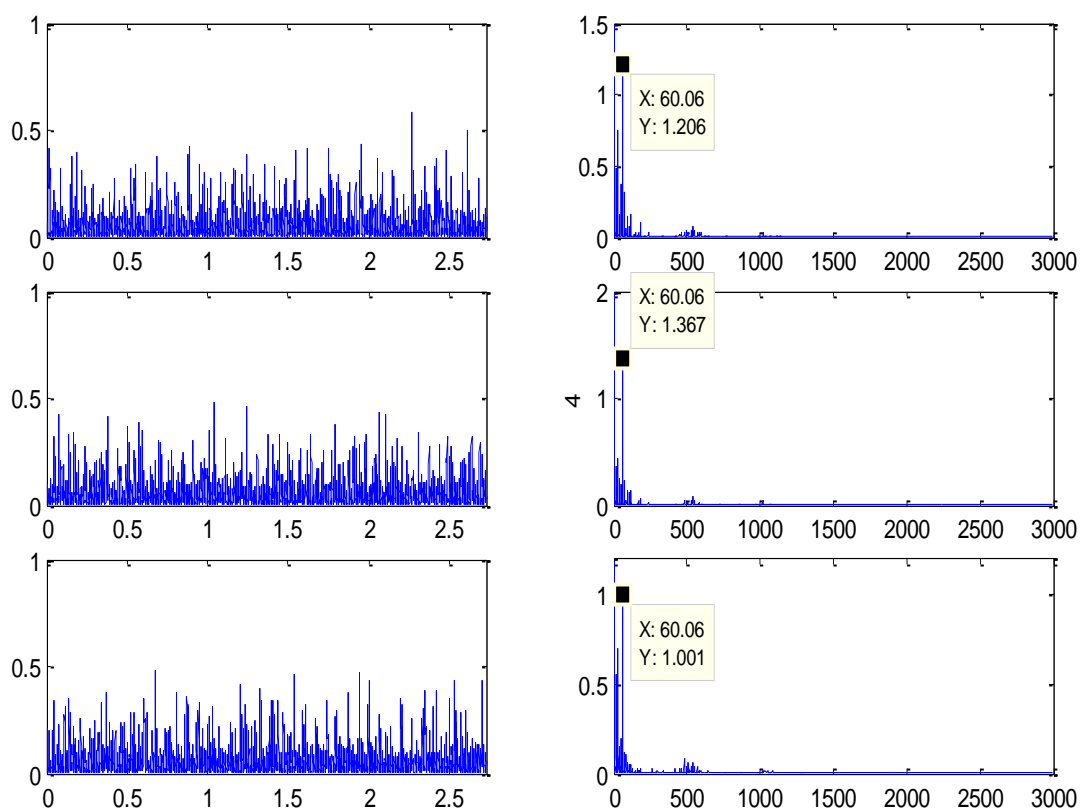


Figura 4.19: Señales de eje sano con mucho ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo (izquierda) y en el dominio de la frecuencia (derecha).

La ventaja de este filtrado es que los valores de los picos de frecuencia de las tres señales más representativas coinciden totalmente (60,06 Hz), al contrario de lo que ocurría con el filtro Smooth, en el que los valores se asemejaban pero variaban ligeramente. El problema es que los valores de los picos son distintos para la señal con ruido y la señal sin ruido, cuando ambas corresponden al mismo eje sano. Es

4. Procesamiento de las señales y análisis preliminares

necesario continuar con el estudio antes de intentar calificar este filtro, con el fin de ver qué ocurre con las señales de eje con defecto.

En la figura 4.20 se muestra la señal de eje con defecto con poco ruido filtrada mediante la Transformada de Hilbert.

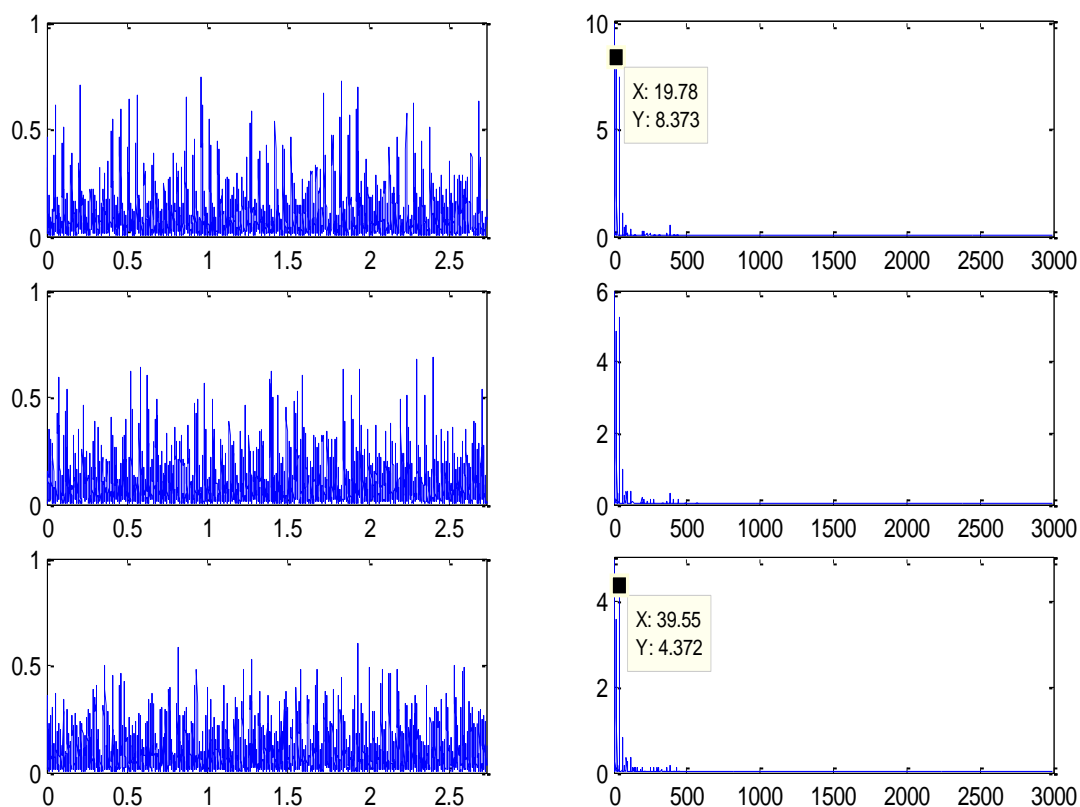


Figura 4.20: Señales de eje con defecto con poco ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo (izquierda) y en el dominio de la frecuencia (derecha).

Los picos de frecuencia vuelven a coincidir en las tres señales representativas y aunque no se vea bien en la figura 4.20, uno de los picos está pegado al eje de ordenadas. El gran problema en este caso es que las frecuencias (19,78 Hz y 39,55Hz) coinciden con las de las señales de eje sano limpias. Esto no interesa en ningún caso, ya que tras el filtrado las señales de eje sano y las de eje con defecto son iguales. Al diagnosticar ambos ejes se obtendrá el mismo resultado cuando debería ser distinto.

4. Procesamiento de las señales y análisis preliminares

Con el fin de comparar en detalle, se representan en la figura 4.21 la FFT de la señal de eje sano limpia (arriba) y la de eje con defecto limpia (abajo), tras ser filtradas mediante la Transformada de Hilbert.

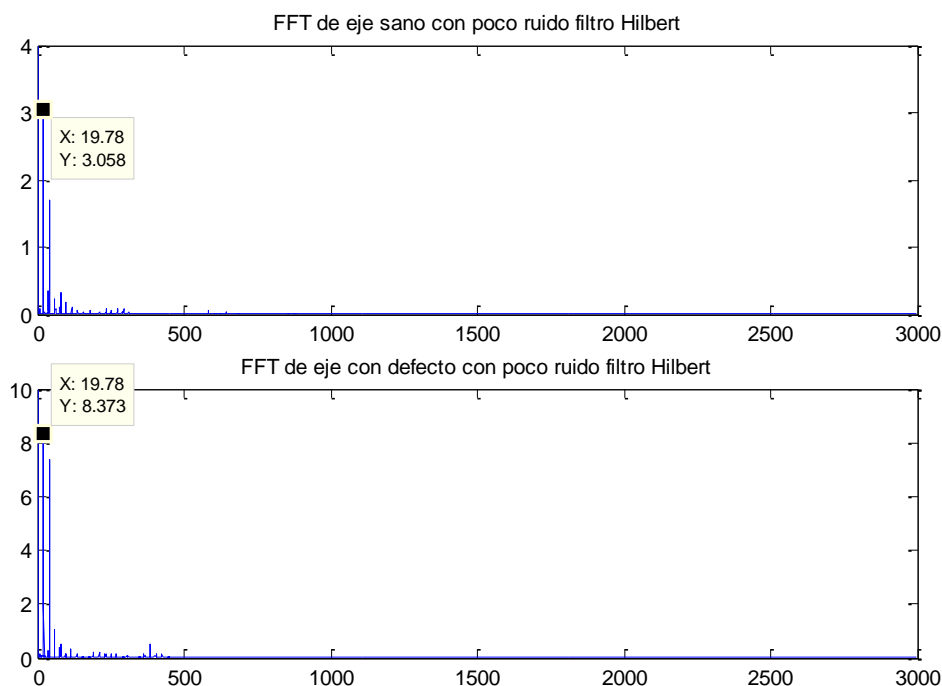


Figura 4.21: Comparación de las FFT de las señales con poco ruido de eje sano (arriba) y con defecto (abajo) procesadas mediante el filtro Hilbert.

Ambas gráficas son prácticamente iguales, lo que dificultará mucho el diagnóstico, ya que éste debe ser distinto. La causa más probable de este fenómeno es que la Transformada de Hilbert elimine la información del defecto considerándola ruido aleatorio. Por tanto, las frecuencias características del defecto desaparecen y la gráfica tiene la misma forma que la del eje sano.

4. Procesamiento de las señales y análisis preliminares

Para terminar el estudio de este filtro, se muestra en la figura 4.22 la señal de eje con defecto y gran presencia de ruido filtrada mediante Hilbert.

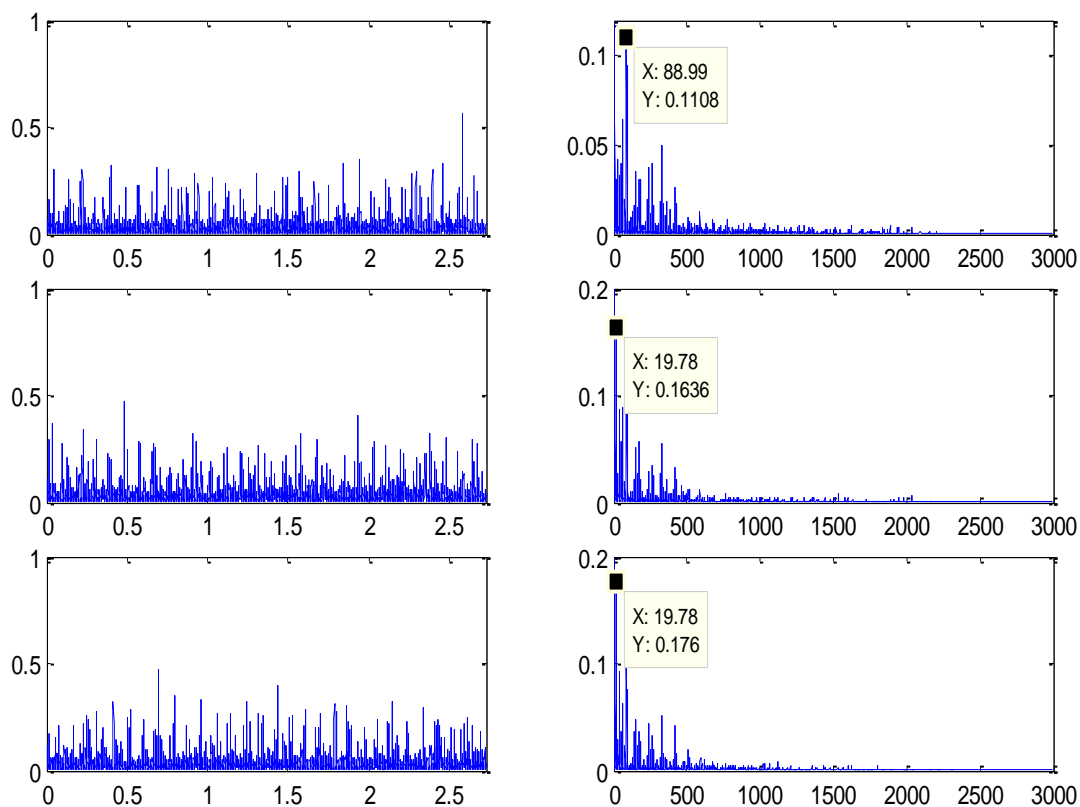


Figura 4.22: Señales de eje con defecto con mucho ruido procesadas mediante el filtro Hilbert, en el dominio del tiempo (izquierda) y en el dominio de la frecuencia (derecha).

En esta ocasión, los picos de frecuencia no coinciden entre las tres señales representativas, además de que la primera sigue presentando mucho ruido. Una vez más, el máximo pico de frecuencia de algunas de las señales se encuentra en 19,78 Hz, exactamente el mismo que el presentado por el eje sano. Por tanto, vuelve a haber confusión en las gráficas entre eje sano y eje con defecto.

Las conclusiones del análisis preliminar son que el filtrado mediante la Transformada Hilbert no va a ser adecuado, ya que las señales de eje sano y las de eje con defecto presentan las mismas características tras ser procesadas. Se cree que el filtro Hilbert, al intentar eliminar todo el ruido posible, acaba con la información del defecto y no permitirá un diagnóstico correcto. A medida que continúe el estudio veremos la validez de este primer análisis.

4.6 Filtrado mediante la Transformada Wavelet (TW)

En este capítulo se va a ver el efecto de la Transformada Wavelet sobre las señales de vibración seleccionadas y se realizará un estudio preliminar de su validez como filtro para el diagnóstico de ejes.

Se han probado distintos tipos de TW y se han ido variando todos los parámetros de éstas, pero no se apreciaban diferencias en los resultados. Primero se selecciona el filtrado por Transformada de Wavelet Estacionaria 1-D y a continuación es necesario especificar el tipo de transformada, el número de niveles de descomposición, el método de umbralización y la estructura del ruido.

Finalmente, se ha decidido utilizar la TW daubechies 6, ya que es una de las más usadas en defectología por sus buenos resultados. Se ha configurado la TW para que tenga 3 niveles de descomposición, umbrales de forma fija blandos y con el ruido blanco sin escalar.

Para comenzar, se filtran las tres señales de eje sano limpias mediante la Transformada Wavelet y el resultado se muestra en la figura 4.23.

4. Procesamiento de las señales y análisis preliminares

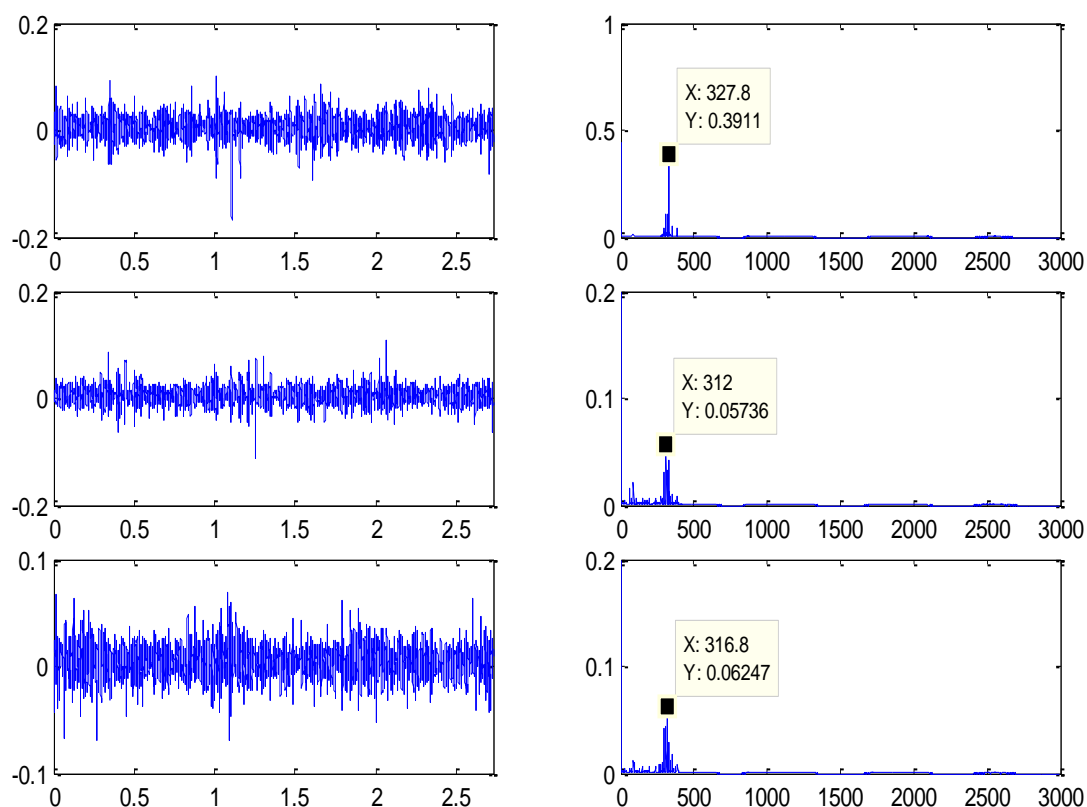


Figura 4.23: Señales de eje sano con poco ruido procesadas mediante el filtro Wavelet, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

Los resultados de la FFT tras el filtrado mediante Wavelet no son tan exactos como en el filtrado Hilbert. Aunque el valor de la frecuencia del pico máximo varía ligeramente de unas señales a otras, el pico destaca de forma importante frente al resto de la gráfica, lo cual es una ventaja. Los valores de la frecuencia pico son 327,8 Hz, 312 Hz y 316,8 Hz por lo que las variaciones no son muy grandes.

4. Procesamiento de las señales y análisis preliminares

Siguiendo con el estudio de este filtro, se representan en la figura 4.24 las señales de eje sano con ruido procesadas mediante la TW.

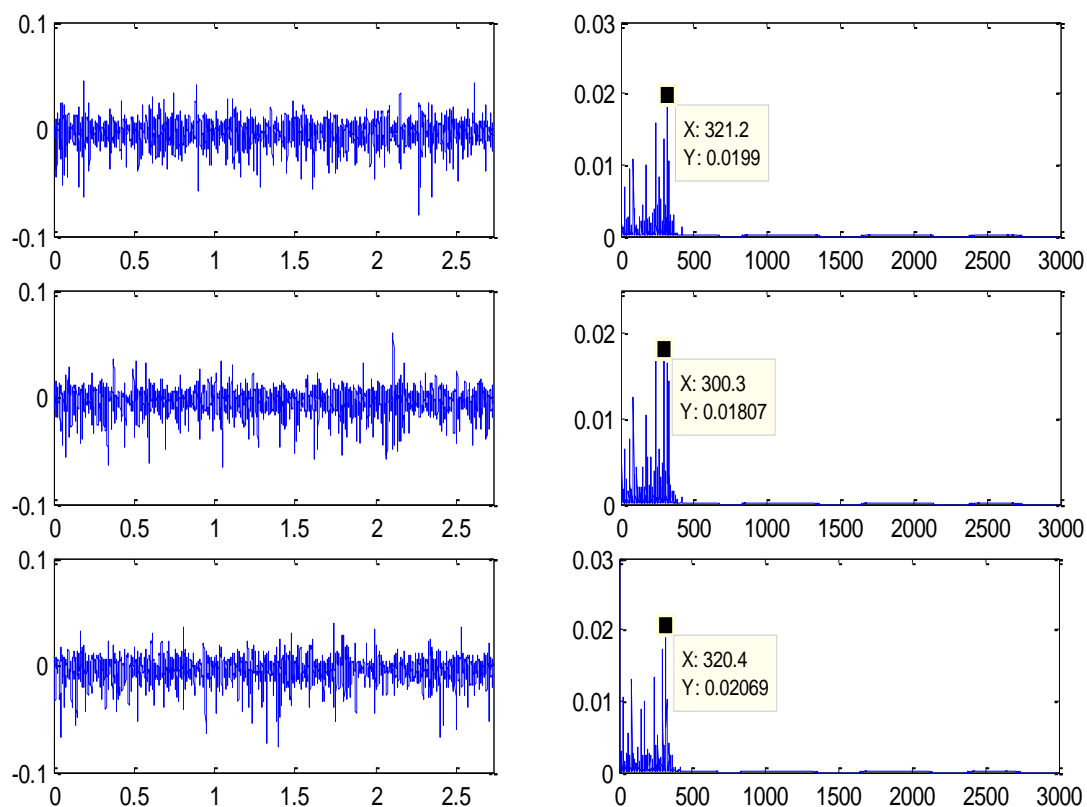


Figura 4.24: Señales de eje sano con mucho ruido procesadas mediante el filtro Wavelet, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

Se observa en la señal una gran presencia de picos de frecuencia, con lo que la eliminación del ruido no parece óptima. La gran ventaja es que el pico máximo de las señales varía poco de unas a otras, con unos valores de 321,2 Hz, 300,3 Hz y 320,4 Hz, y se parece muchísimo al obtenido en las señales de eje sano con poco ruido, cuyos valores rondaban los 320 Hz. Por tanto, a pesar de la presencia de ruido en la señal, se ha conseguido destacar la frecuencia característica de eje sano, lo cual podría facilitar enormemente el diagnóstico del mismo.

4. Procesamiento de las señales y análisis preliminares

Se muestra la figura 4.25 con el fin de poder comparar la FFT de la señal filtrada mediante TW de eje sano con poco ruido (arriba) y con mucho ruido (abajo).

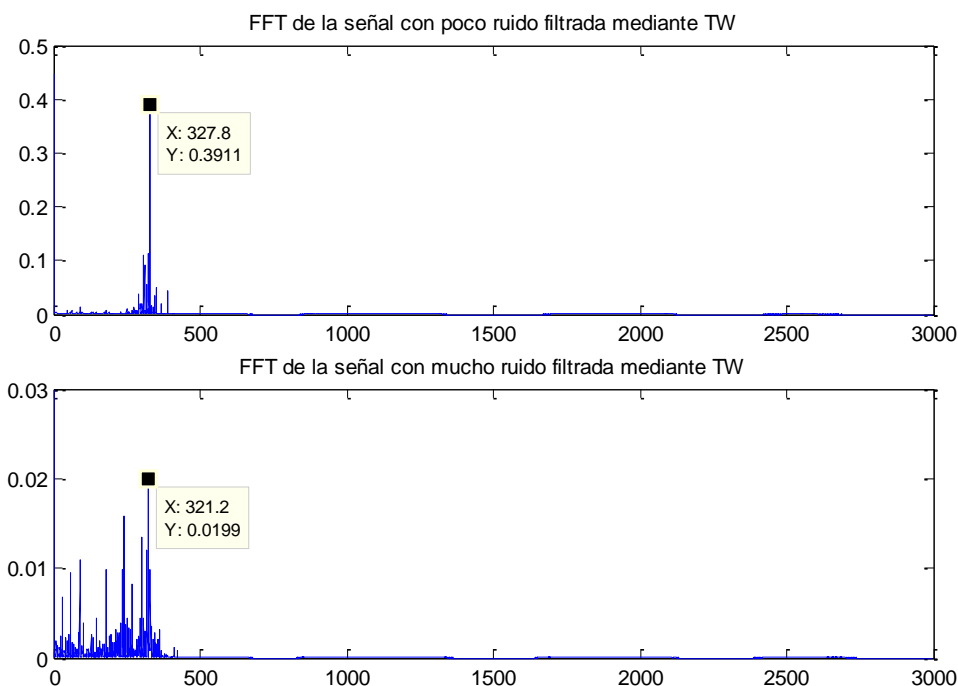


Figura 4.25: Comparación de la FFT de la señal de eje sano con poco ruido (arriba) y con mucho ruido (abajo) filtradas mediante TW.

Se ve que, a pesar del filtrado, hay más ruido en la señal que inicialmente estaba más sucia, pero el pico máximo tiene el mismo valor de frecuencia en ambas señales. Por el momento, las primeras impresiones del filtrado mediante la TW son bastante positivas.

4. Procesamiento de las señales y análisis preliminares

Pasando a la segunda parte del estudio de este filtro, se muestran en la figura 4.26 las señales de eje con defecto procesadas con la TW.

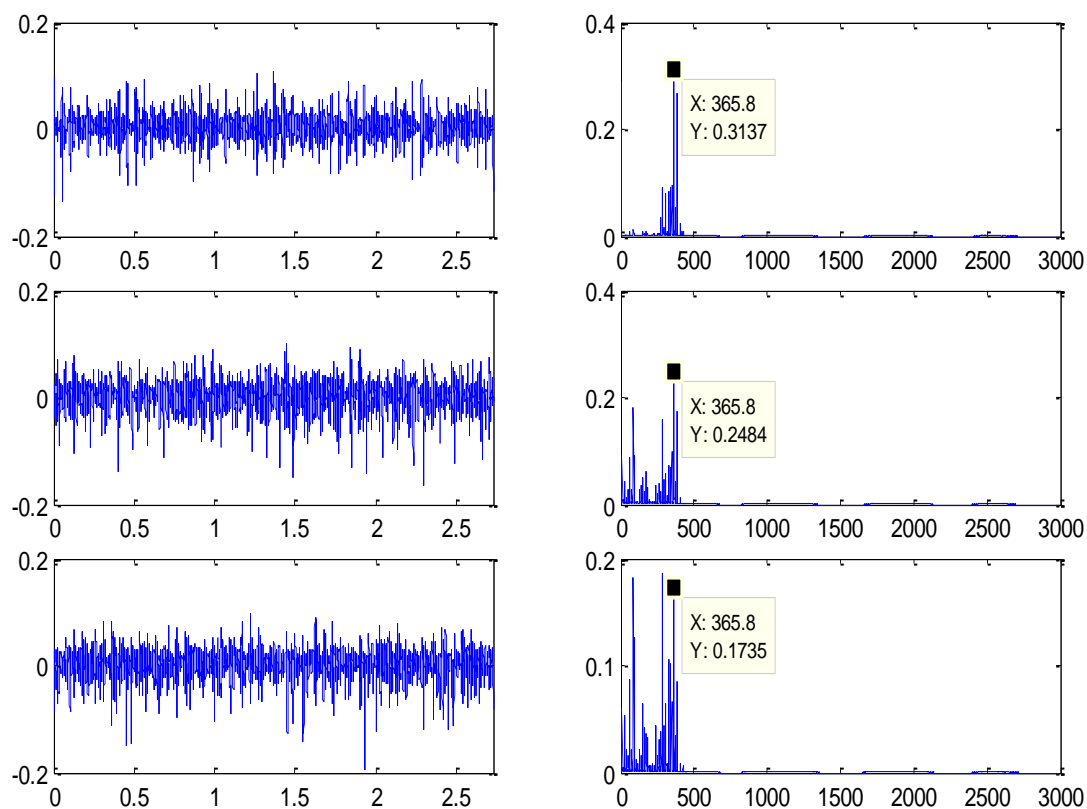


Figura 4.26: Señales de eje con defecto con poco ruido procesadas mediante el filtro Wavelet, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

Es interesante ver que esta vez, la frecuencia 365,8 Hz se repite con exactitud en la FFT de las tres señales más representativas. Es el pico máximo en las dos primeras y uno de los mayores en la tercera. A simple vista parece que ésta será la frecuencia característica del defecto y queda por ver si también destaca en las señales con gran presencia de ruido.

Dicho esto, se procede a analizar las señales de eje con defecto que tienen mucho ruido filtradas mediante TW, mostradas en la figura 4.27.

4. Procesamiento de las señales y análisis preliminares

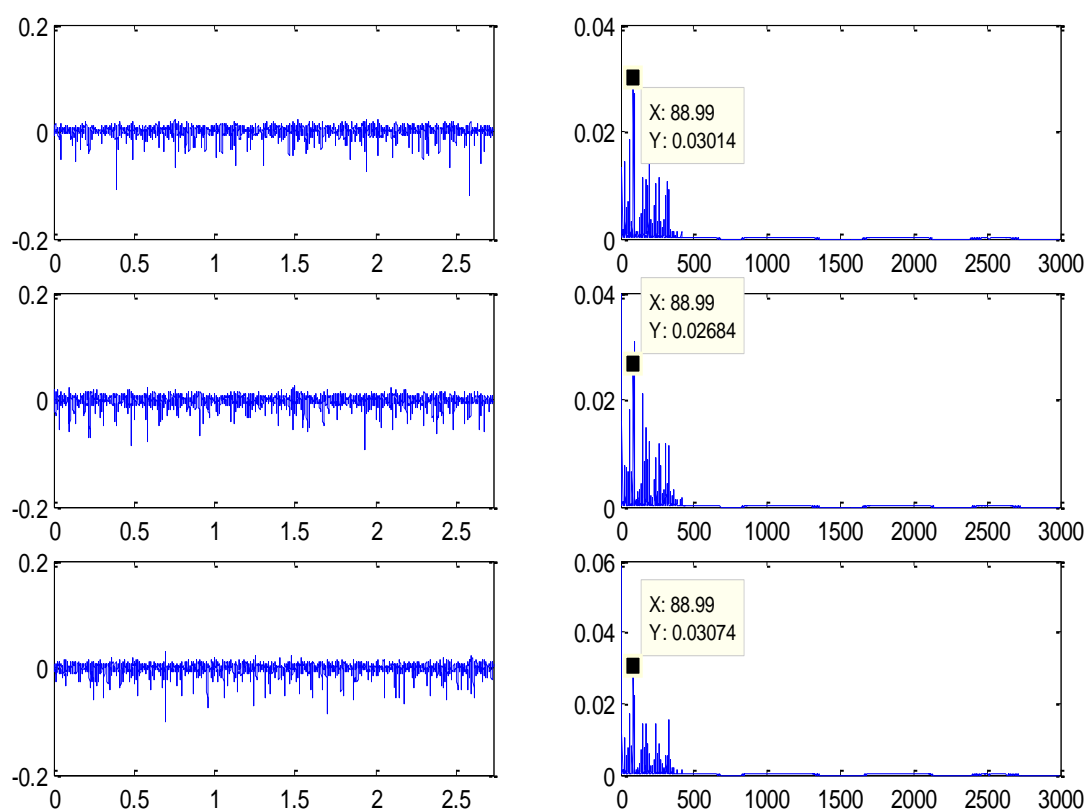


Figura 4.27: Señales de eje con defecto con mucho ruido procesadas mediante el filtro Wavelet, en el dominio temporal (izquierda) y en el dominio de la frecuencia (derecha).

En esta ocasión, es la frecuencia 88,99 Hz la que se repite en la FFT de las tres señales. Es el mayor pico en la primera y tercera, y el segundo mayor en la segunda. Esta frecuencia dista mucho de la obtenida para las señales con poco ruido (365,8 Hz), las cuales se consideran más fiables a la hora de identificar el defecto. La causa más probable es que alguno de los ruidos haya eclipsado la información del defecto y del eje, con lo que en este caso el filtrado no habría sido efectivo. También se puede observar que las tres señales temporales tienen un patrón muy parecido.

La primera conclusión sacada de este pequeño estudio preliminar es que el filtrado mediante TW es muy efectivo para las señales de eje sano, ya que aísla muy bien la frecuencia característica del eje, eliminando una gran cantidad de ruido. El problema viene al utilizar este filtro para diagnosticar ejes con defecto, en cuyo caso la TW destaca la frecuencia de algunos ruidos eliminando la del eje y el defecto. Todos estos análisis preliminares se podrán confirmar o desmentir mediante el uso de redes neuronales artificiales.

5. REDES NEURONALES ARTIFICIALES

5.1 Introducción

Las redes neuronales artificiales se utilizarán para poner a prueba los análisis preliminares realizados anteriormente. Aunque se han sacado conclusiones observando las distintas señales filtradas, se desconocen los valores de la frecuencia característica del eje sano y del eje con defecto. Por tanto, no podemos comparar los picos de las señales obtenidas con ningún valor cuya validez haya sido demostrada.

Por este motivo, se usan las redes neuronales para comparar la eficacia de cada filtrado para el diagnóstico de ejes. Para ello se usará la red neuronal de reconocimiento de patrones (Pattern Recognition Network), preprogramada en Matlab®. Esta red es de tipo "feedforward" (proalimentación) y puede ser entrenada para clasificar las entradas conforme a las clases objetivo. En este caso, las entradas son las señales de vibración y las clases objetivo son eje sano o eje con defecto. La red se encargará de la búsqueda de patrones en las señales de entrenamiento, tanto de eje sano como de eje con defecto, con el fin de determinar posteriormente si las señales de test pertenecen a una clase u otra. La validez de cada filtro se determinará a partir de la tasa de acierto de la red al diagnosticar a qué tipo de eje corresponden las señales. Tasas de acierto mayores suponen filtrados más adecuados.

En esta parte del estudio, se utilizarán todas las señales disponibles medidas para eje sano y para eje con defecto al 100% del radio, con el fin de mejorar el entrenamiento de la red neuronal y mejorar la precisión del diagnóstico.

Para programar la red neuronal, es necesario especificar ciertos parámetros, como el número de neuronas de la capa oculta o los porcentajes de los datos de entrada a la red que se usarán para el entrenamiento, para la validación y para el test. Los datos de entrenamiento se utilizan para enseñar a la red a clasificar correctamente las señales de entrada. Los datos de validación sirven para determinar si el entrenamiento de la red es adecuado o no, y se utilizan como criterio de parada. Por último, los datos

5. Redes neuronales artificiales

de test se usan para poner a prueba la red neuronal. . Al entrenar la red se realiza la validación y el test en cada iteración.

En este proyecto se han usado 8 neuronas en la capa oculta de la red y los porcentajes de datos dedicados a entrenamiento, validación y test varían para cada caso de estudio. El proceso de entrenamiento parará cuando el error de validación aumente durante 100 iteraciones seguidas. Además, se da a la red un límite de 10000 iteraciones, al que no debería llegar nunca.

El estudio de cada filtro se va a dividir en dos partes:

- En la primera, se creará la red neuronal a partir de todas las carpetas de señales disponibles. La red toma el 60% de los datos para el entrenamiento, el 20% para validación y el 20% restante para test, que son los valores que vienen por defecto en las redes neuronales de reconocimiento de patrones. Se toma como porcentaje de acierto el valor proporcionado por la propia red al realizar el test.
- En la segunda se tomarán solamente el 57% de las carpetas (8 de 14 en eje sano y 4 de 7 en eje con defecto) de señales para crear la red. De estas carpetas seleccionadas se tomará el 98% de los datos para entrenamiento y tan sólo el 1% para validación y el 1% para test. En este caso, el test realizado por la propia red durante el proceso de entrenamiento no se tendrá en cuenta. En cambio, una vez creada la red, se introducirán los datos de las carpetas restantes para su clasificación y se determinará el porcentaje de señales clasificadas correctamente.

En principio, estos dos métodos pueden parecer iguales pero hay claras diferencias. Las 100 señales de cada carpeta son muy parecidas, ya que se miden todas seguidas, con tan solo un segundo de margen entre cada señal y la siguiente. Por tanto, las condiciones de ruido son muy parecidas. Al entrenar, validar y testear la red con todas las carpetas, en el primer caso, las señales de entrenamiento y las de test será muy parecidas entre sí, por lo que se esperan unas tasas de acierto muy elevadas. En cambio, en el segundo caso, las carpetas de entrenamiento y de test serán distintas. Por este motivo, las señales de entrenamiento y de test pueden tener niveles totalmente distintos de ruido, con lo que se pondrá verdaderamente a prueba la validez

5. Redes neuronales artificiales

de los filtros utilizados. En este método de estudio se esperan tasas de acierto mucho menores.

El inconveniente del uso de redes neuronales es que la tarea de escoger qué datos se utilizan para entrenamiento, validación y test es aleatoria (siempre adecuándose a los porcentajes especificados). Por tanto, cada vez que se realiza el entrenamiento se escogen unos datos distintos para cada parte. Esto lleva a una ligera variación de los resultados cada vez que se ejecuta el programa de entrenamiento. Para que los resultados del proyecto tengan la máxima precisión posible, se han realizado varios entrenamientos para cada tipo de filtro y se ha utilizado finalmente la red que proporcionara los resultados que más veces se han repetido.

De todas formas, para utilizar las redes neuronales artificiales es necesaria la previa descomposición de la señal en paquetes de energía, por su mayor interés para la defectología.

5.2 Paquetes de energía

La descomposición de la señal en paquetes de energía se realiza mediante una Transformada Wavelet Packets. La TWP escogida es de tipo daubechies 6 con 3 niveles de descomposición.

Una vez descompuesta la señal, se hallan los porcentajes de energía correspondientes a los nodos terminales del árbol de descomposición Wavelet (recordemos la figura 2.5). Estos porcentajes son los paquetes de energía de cada señal, que se introducirán en la red neuronal en lugar de usar directamente los datos medidos o filtrados.

Tanto la descomposición de la señal mediante TWP como la obtención de los porcentajes de energía se han realizado mediante algoritmos en Matlab®. Se realizan tras el filtrado de la señal.

5.3 Entrenamiento y test con todas las carpetas

Como ya se ha explicado antes, en esta parte del estudio se utilizan todas las carpetas tanto para el entrenamiento como para la validación y el test de la red neuronal. De esta forma, la red neuronal coge todos los datos y utiliza un 60% aleatorio para el entrenamiento de la red, un 20% para la validación y el 20% restante para el test, que determina el porcentaje de acierto y por tanto la efectividad de dicha red. Por tanto se utilizarán datos parecidos para el entrenamiento y el test. De antemano se esperan tasas de acierto muy altas.

Se creará una red neuronal para cada tipo de filtro. Primero se filtran las señales y posteriormente se utilizan para el entrenamiento, validación y test de la red. A continuación se mostrarán los resultados de cada red neuronal dependiendo del filtro utilizado.

5.3.1 Señales originales sin filtrar

En primer lugar se va a estudiar la eficacia que alcanza la red neuronal de reconocimiento de patrones para las señales sin filtrar, con el fin de poder comparar los resultados con los de las señales filtradas.

Cada vez que se entrena una red neuronal se obtienen resultados ligeramente distintos, ya que los datos escogidos para entrenamiento, validación y test son aleatorios y, por tanto, diferentes cada vez. Tras hacer varios entrenamientos, se escogerán las redes cuyos valores sean intermedios.

Tras programar la red neuronal, se utiliza su toolbox de Matlab® para observar los resultados. La palabra toolbox se utiliza en Matlab® para designar a las interfaces gráficas que se utilizan para controlar ciertos procesos. En la figura 5.1 se muestra la toolbox de la red neuronal para señales sin filtrar.

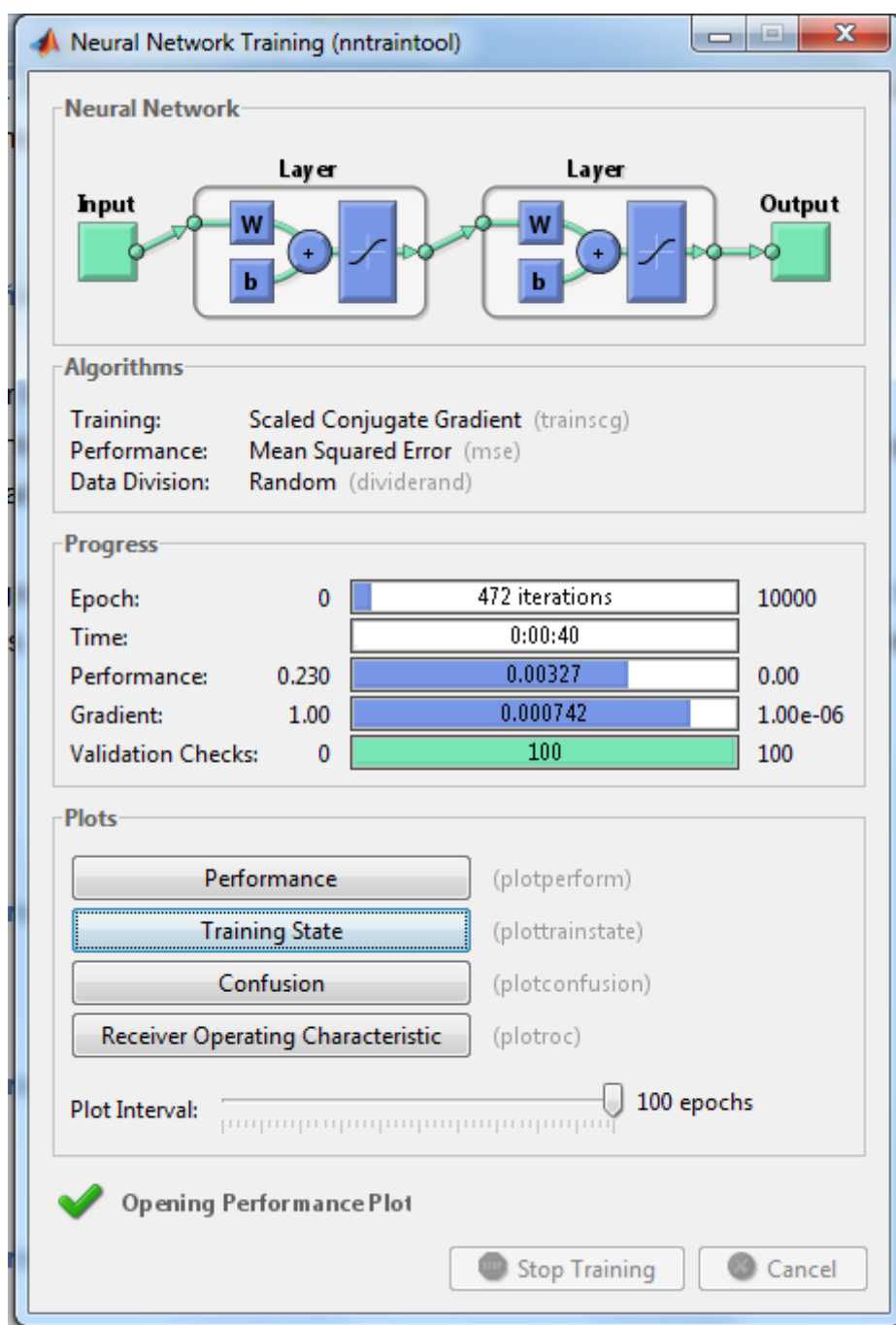


Figura 5.1: Toolbox de la red neuronal para señales sin filtrar (caso 1).

Se puede observar que el entrenamiento se ha realizado en 472 iteraciones y 40 segundos, que el rendimiento del entrenamiento alcanza un valor de 0.00327 y el gradiente 0,000742. Cuanto menores sean los valores del rendimiento y el gradiente, mayor precisión alcanzará la red neuronal, ya que durante el entrenamiento se busca minimizar estos valores.

5. Redes neuronales artificiales

A continuación se van a analizar los distintos gráficos de la red neuronal. El primero de ellos es el del Rendimiento durante el entrenamiento (figura 5.2), que muestra la tendencia del error cuadrático medio del entrenamiento, validación y test a medida que pasan las iteraciones.

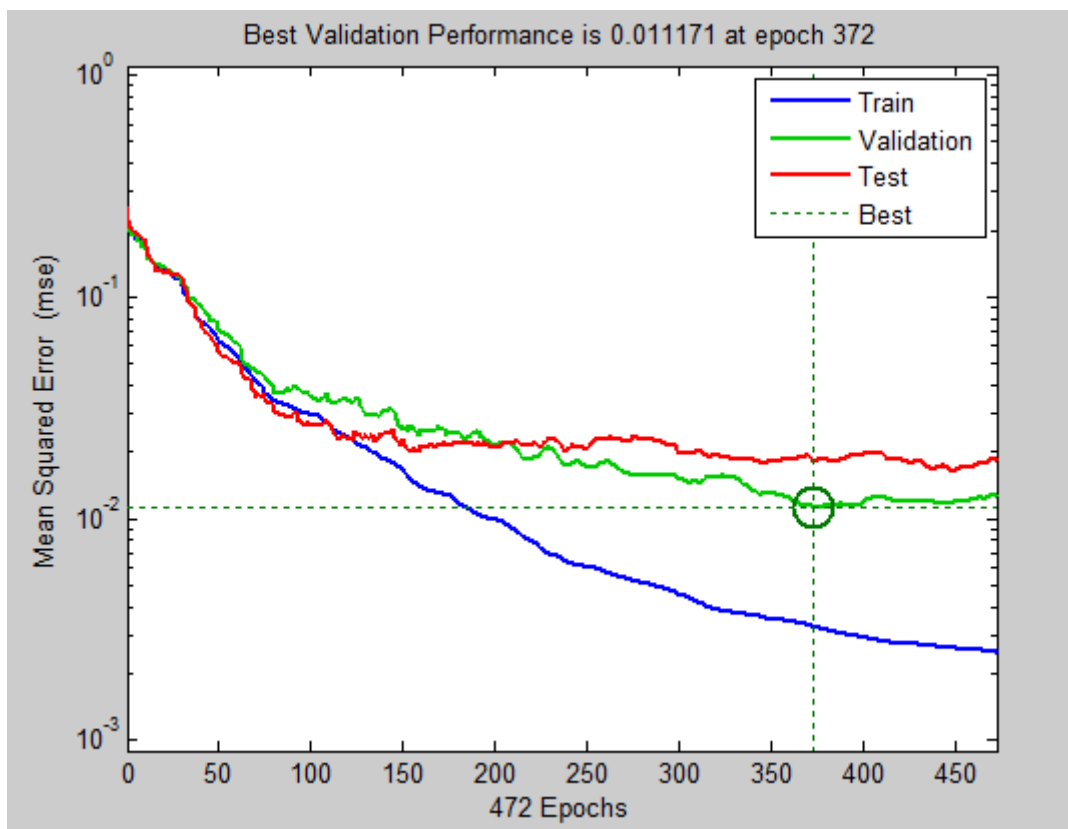


Figura 5.2: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales sin filtrar (caso 1).

Se ve como el valor de los errores de los tres procesos van bajando hasta que en la iteración 372, el error de la validación alcanza su valor mínimo: 0,011171. El error del entrenamiento sigue bajando hasta el valor señalado en la toolbox (0,00327), pero el entrenamiento se detiene 100 iteraciones después porque el error de la validación no disminuye más. Recordemos que este es el criterio de parada del entrenamiento de la red.

El siguiente gráfico que se va a analizar es el de Estado de Entrenamiento, mostrado en la figura 5.3. Éste se compone de dos diagramas: el primero muestra la tendencia del gradiente con el paso de las iteraciones y el segundo el número de iteraciones seguidas en las que el error de la validación ha ido aumentando. Este segundo

5. Redes neuronales artificiales

diagrama simplemente va creciendo cuando se van dando valores mayores en el error de la validación que el mínimo alcanzado hasta ese momento en el entrenamiento. Si en una determinada iteración el valor del error vuelve a disminuir, la gráfica vuelve a cero. Por este motivo, al final de la gráfica el valor crece hasta 100, porque es el número de iteraciones que hace detener el entrenamiento.

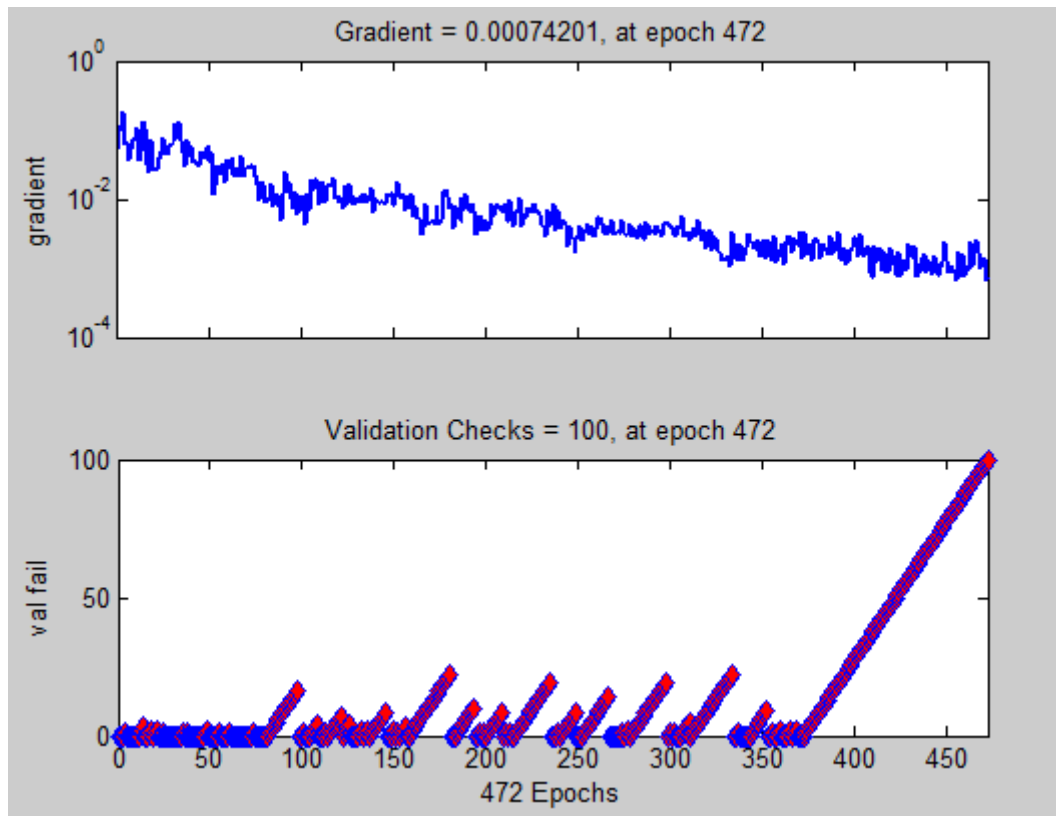


Figura 5.3: Gráficos del Estado de Entrenamiento de la red neuronal para señales sin filtrar (caso 1).

Se puede observar que el gradiente siempre tiene una tendencia decreciente a medida que van pasando las iteraciones, aunque el decrecimiento se va volviendo muy sutil al final del entrenamiento.

El tercer diagrama que ofrece la toolbox es el de la Confusion (figura 5.4), que nos permite analizar la respuesta de la red neuronal, comparando las salidas de la red entrenada con los resultados esperados. La primera matriz nos muestra los datos del entrenamiento, la segunda los de la validación, la tercera los del test y la última junta todos los datos.

5. Redes neuronales artificiales

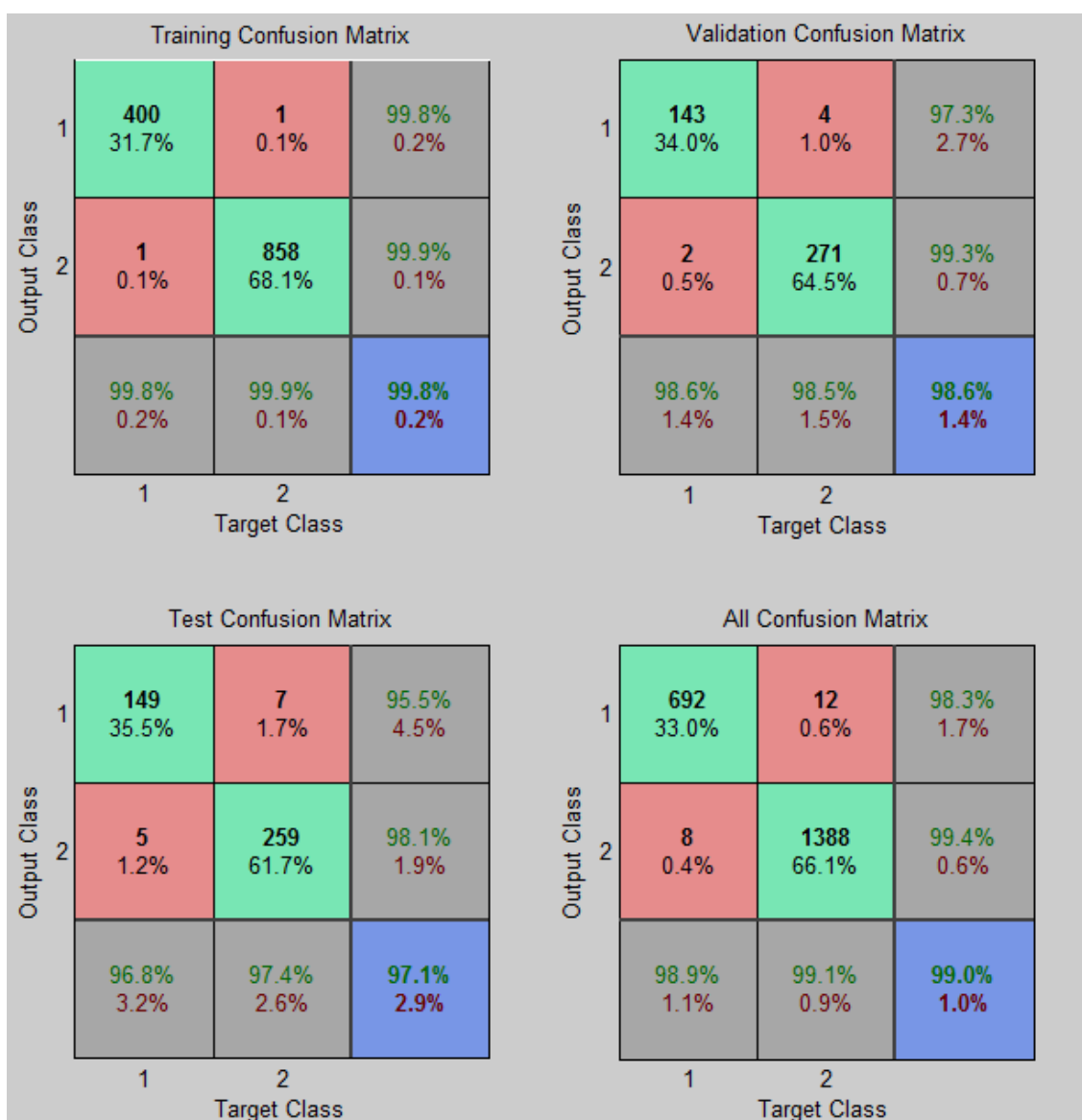


Figura 5.4: Matrices de confusión de la red neuronal para señales sin filtrar (caso 1).

La lectura de las matrices se realiza por columnas. En la primera columna se observan las señales de eje con defecto, las que fueron clasificados correctamente en verde y las que se clasificaron de forma incorrecta en rojo. Por tanto, en esta columna, las que se muestran en color rojo fueron diagnosticadas por la red neuronal como eje sano cuando en realidad pertenecían a eje con defecto. Al final de la columna se muestra el porcentaje de señales de eje con defecto diagnosticadas correctamente (en verde) e incorrectamente (en rojo).

En la segunda columna se muestran las señales de eje sano. Una vez más los cuadros verdes corresponden a las diagnosticadas correctamente y los rojos a las

5. Redes neuronales artificiales

diagnosticadas incorrectamente. El cuadro gris al final de la columna vuelve a señalar el porcentaje de señales correcta e incorrectamente diagnosticadas, esta vez de eje sano.

Finalmente, el cuadro azul representa el porcentaje total de señales cuyo diagnóstico ha sido correcto.

Como se ve en la matriz de confusión de todos los datos, se han utilizado un total de 700 señales de eje con defecto y 1400 de eje sano para el estudio de las redes neuronales.

Al analizar la figura 5.4, se puede observar que en las redes neuronales entrenadas con todas las carpetas, el porcentaje de acierto sin filtrar las señales es altísimo. Entre los datos de entrenamiento sólo se han diagnosticado incorrectamente una señal de eje sano y una de eje con defecto, alcanzando un porcentaje de acierto del 99,8%. De todas formas, la matriz que vamos a utilizar para considerar la validez de cada filtro es la de los datos de test. De los datos de eje sano, se han diagnosticado correctamente 259 y sólo se han cometido 7 fallos; mientras que de los de eje sano se ha acertado con 149 y se ha fallado con 5. Por tanto, el porcentaje de acierto total es del 97,1%.

Se puede estipular que al utilizar todas las carpetas, con datos parecidos en el entrenamiento y test, las redes neuronales aciertan casi siempre para señales sin filtrar.

5.3.2 Filtrado Smooth

En esta ocasión, la red neuronal se va a crear con las señales previamente procesadas mediante el filtro Smooth y se van a analizar los resultados obtenidos. Tras entrenar la red se obtiene su toolbox, mostrada en la figura 5.5.

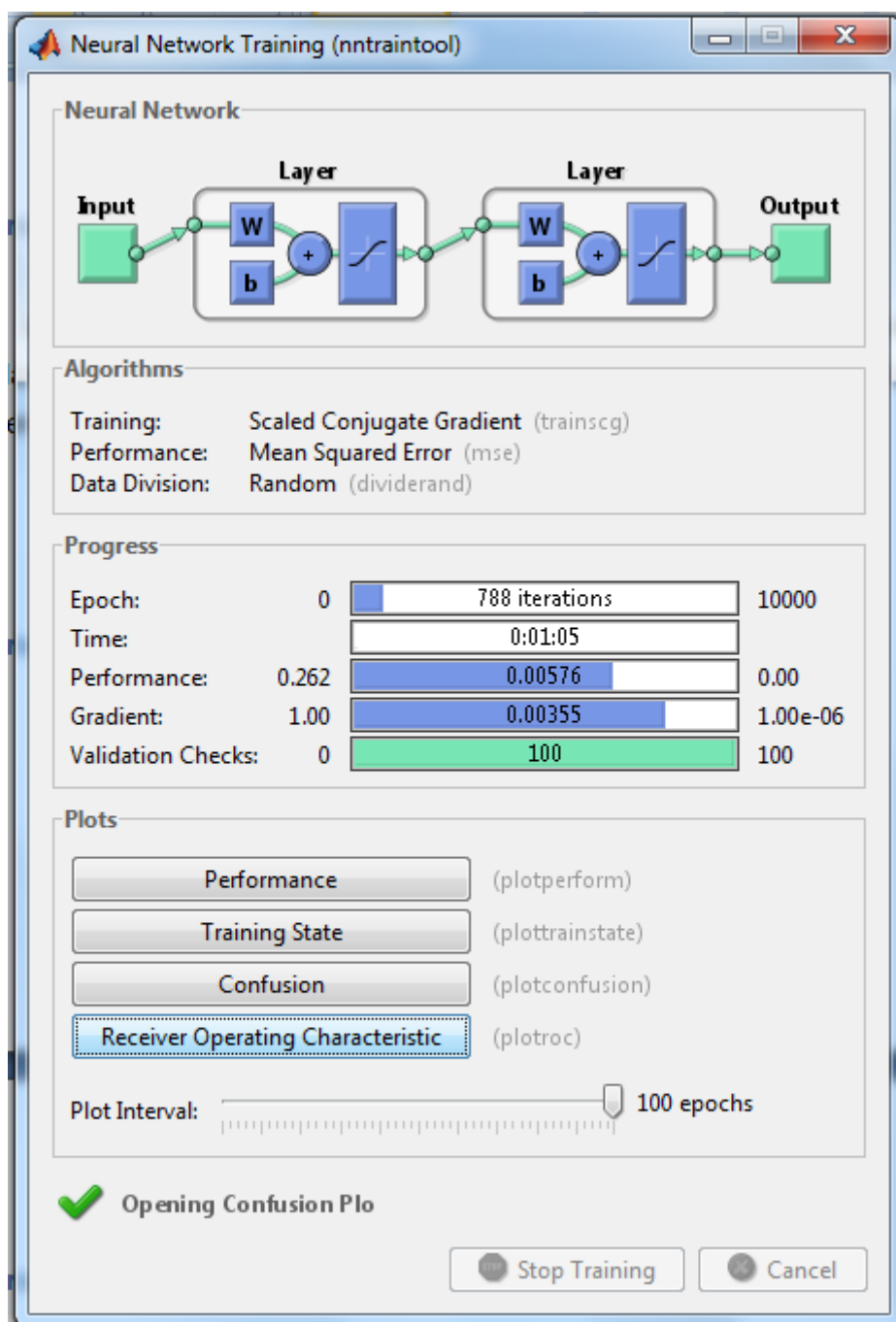


Figura 5.5: Toolbox de la red neuronal para señales procesadas con el filtro Smooth (caso 1).

5. Redes neuronales artificiales

El número de iteraciones necesarias para el entrenamiento asciende a 788 y el tiempo necesario a un minuto y cinco segundos. Por tanto, las señales filtradas han necesitado más tiempo e iteraciones para terminar el entrenamiento de la red neuronal. Además, el rendimiento alcanza un valor de 0.00576 frente al 0.00327 que alcanzaba en el caso anterior. Por tanto, el error cometido por las estimaciones de la

red neuronal será mayor. A su vez, el gradiente tiene un valor de 0.00355, mucho mayor que 0.000742 del caso anterior.

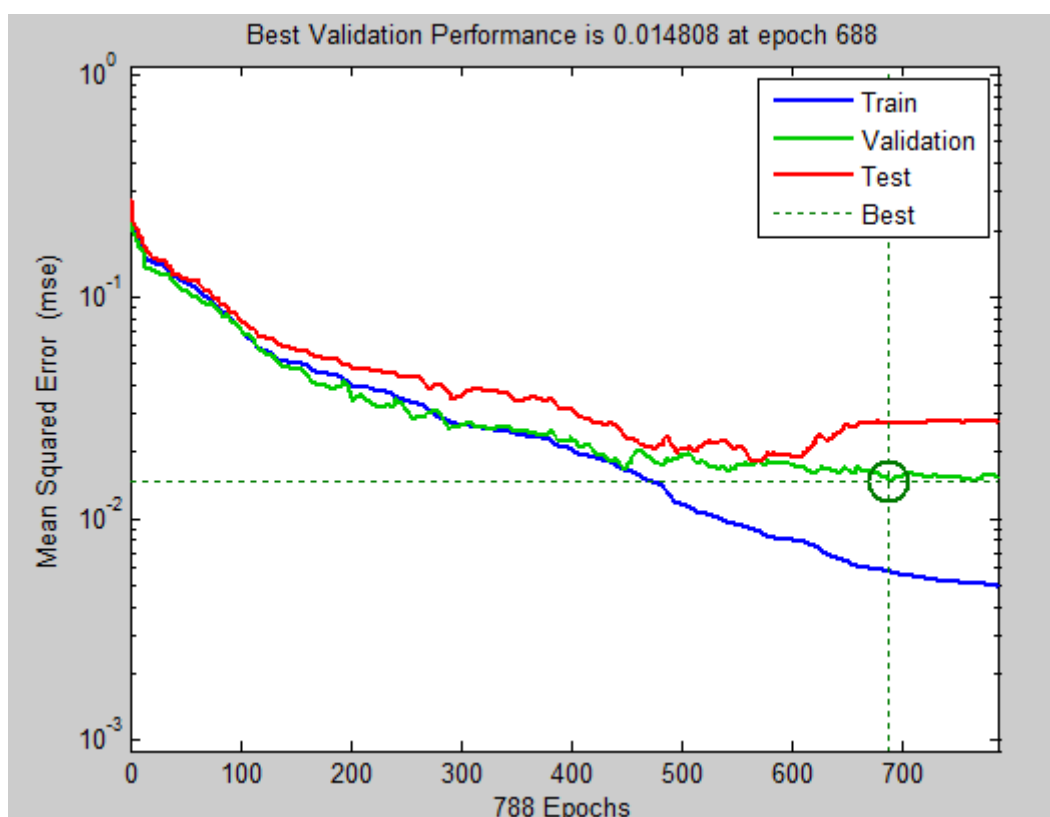


Figura 5.6: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Smooth (caso 1).

La figura 5.6 muestra el gráfico del rendimiento, en el que se puede observar que la tendencia es parecida a la del caso anterior, con los valores menores de error en el entrenamiento, seguido de la validación. El mejor valor del rendimiento para la validación es de 0.014808 en la iteración 688, el cual es un poco peor que en el caso sin filtrar (0.011171).

5. Redes neuronales artificiales

Los datos más importantes para la comparación son los de las matrices de confusión, mostradas en la figura 5.7.

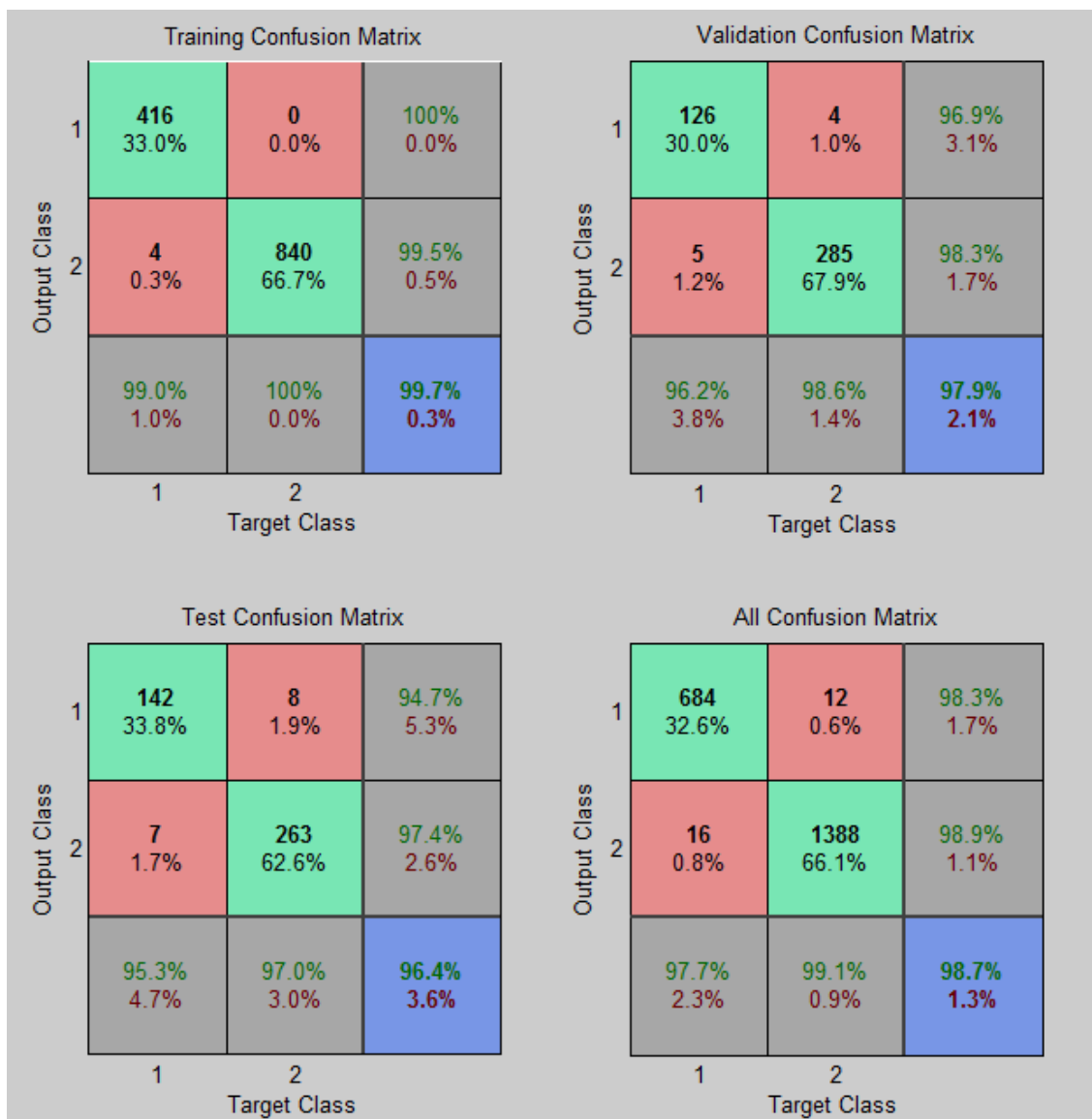


Figura 5.7: Matrices de confusión de la red neuronal para señales procesadas mediante el filtro Smooth (caso 1).

Como se podía predecir mediante el gráfico del rendimiento, la mejor tasa de acierto se ha logrado en el entrenamiento (99,7%), seguida de la validación (97.9%), pero el valor realmente importante es el del test, que es del 96,4%. Sólo se han diagnosticado incorrectamente 7 señales de eje con defecto frente a las 142 acertadas y 8 señales de eje sano frente a las 263 correctas. Por tanto, los valores son ligeramente inferiores a los alcanzados para las señales sin filtrar.

5. Redes neuronales artificiales

Se presenta la tabla 5.1 para poder comparar con más facilidad todos los valores de la red creada a partir de las señales sin filtrar con la creada a partir de las señales procesadas mediante el filtro Smooth.

Tabla 5.1: Comparación de las redes creadas con y sin filtrado de las señales.

	Sin filtrar	Filtro Smooth
Tasa de acierto	97,10%	96,40%
Iteraciones	472	788
Tiempo	40	1:05
Rendimiento	0,00327	0,00576
Gradiente	0,000742	0,00355

Todos los valores son mejores para las señales sin filtrar. El motivo más probable es que al utilizar datos muy parecidos para el entrenamiento y el test, la eliminación de ruido en la señal puede ser desfavorable. Esto se debe a que hay señales con exactamente el mismo nivel de ruido en el entrenamiento y en el test, con lo que son realmente parecidas. En cambio, al eliminar parte del ruido, los patrones de estas señales pueden tener más diferencias.

5.3.3 Filtrado Hilbert

En este capítulo se utilizarán las señales filtradas mediante la Transformada de Hilbert para crear la red neuronal. En la figura 5.8 se puede observar la toolbox de la red, con algunos de los datos de entrenamiento.

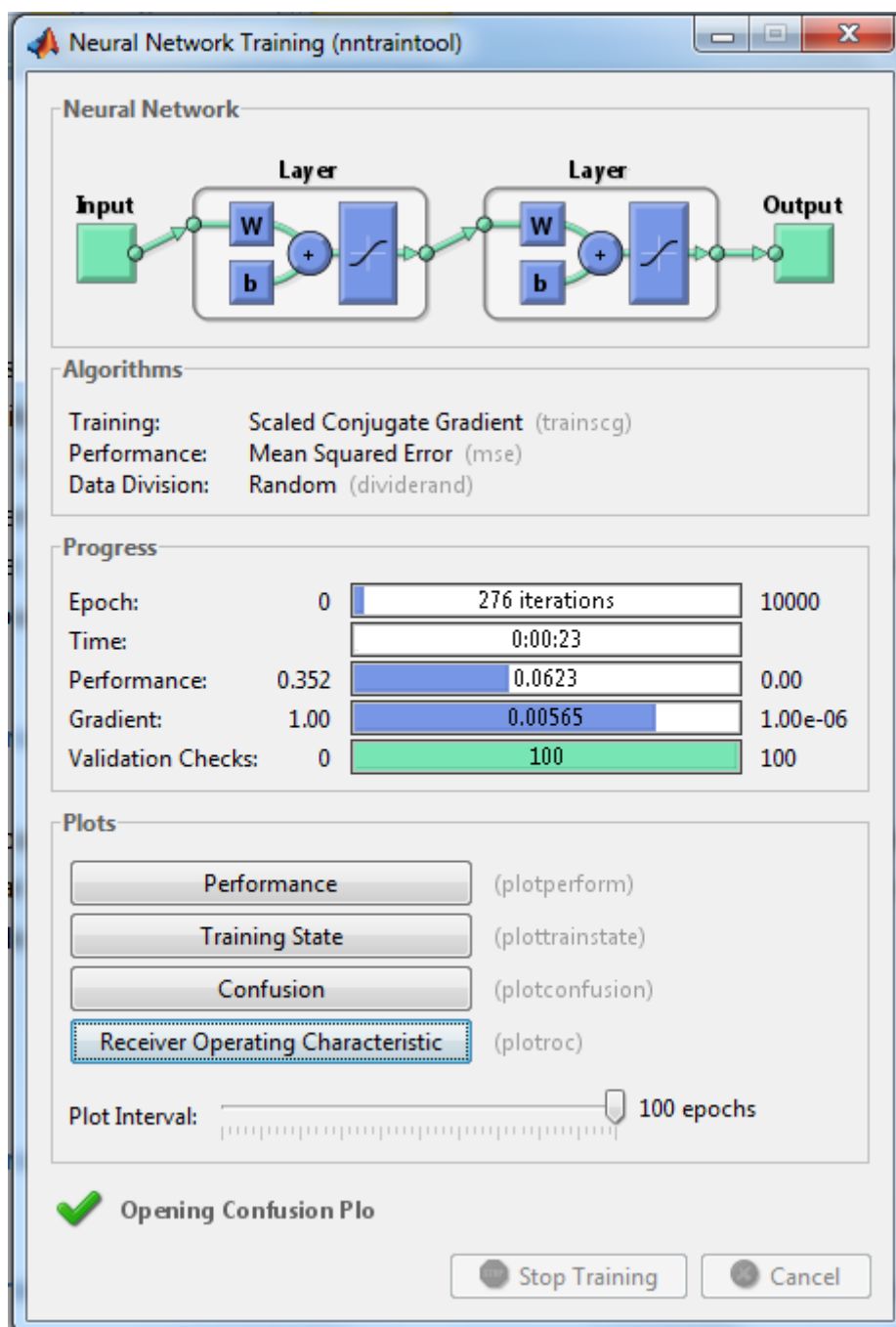


Figura 5.8: Toolbox de la red neuronal para señales procesadas con el filtro Hilbert (caso 1).

5. Redes neuronales artificiales

Se observa que esta vez el entrenamiento se ha realizado en tan sólo 276 iteraciones y 23 segundos. Tanto el rendimiento como el gradiente alcanzado son peores que los de los casos anteriores, con unos valores de 0.0623 y 0.00565 respectivamente.

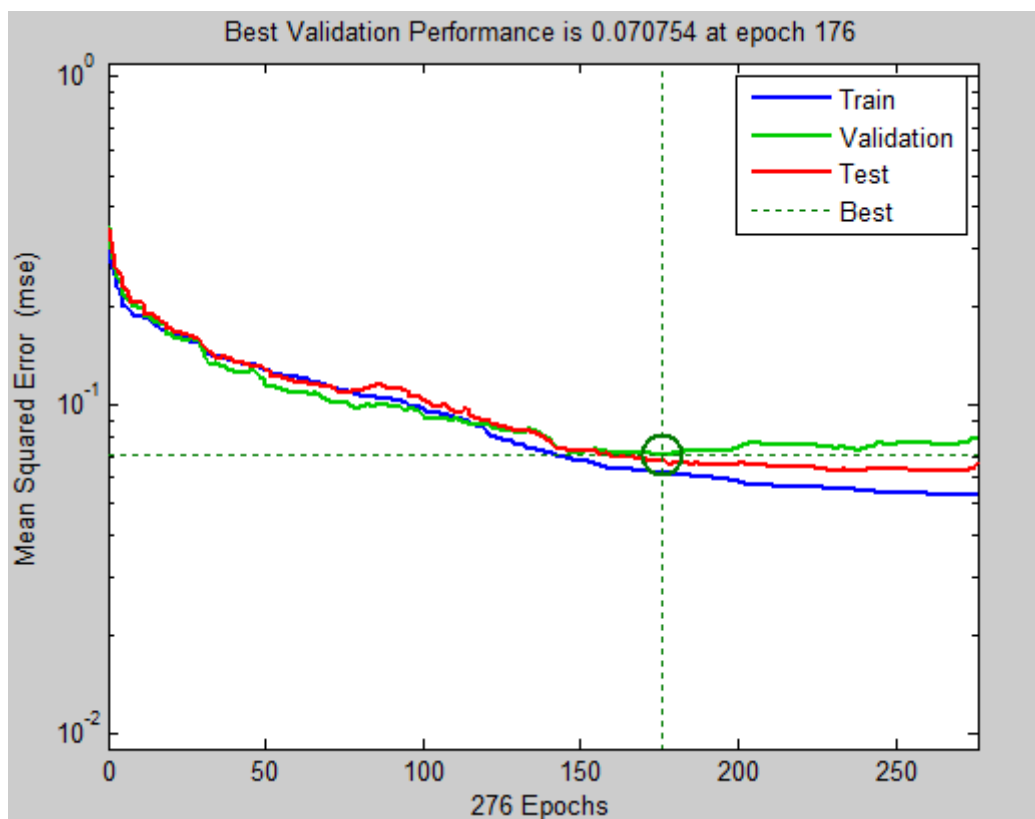


Figura 5.9: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Hilbert (caso 1).

En este caso, la tendencia del error (figura 5.9) varía un poco respecto a los casos anteriores. El error en entrenamiento vuelve a ser el mínimo, pero esta vez le sigue el error del test, siendo el de la validación el mayor de todos. Esto puede ser debido a la pequeña cantidad de iteraciones que se han realizado. La gráfica, además, muestra un decrecimiento más brusco al principio del todo, con una tendencia a estabilizarse desde las 150 iteraciones. El mejor valor del rendimiento es de 0,070754 en la iteración número 176.

5. Redes neuronales artificiales

Una vez más, se acude a las matrices de confusión, mostradas en la figura 5.10, para ver la tasa de acierto de la red.

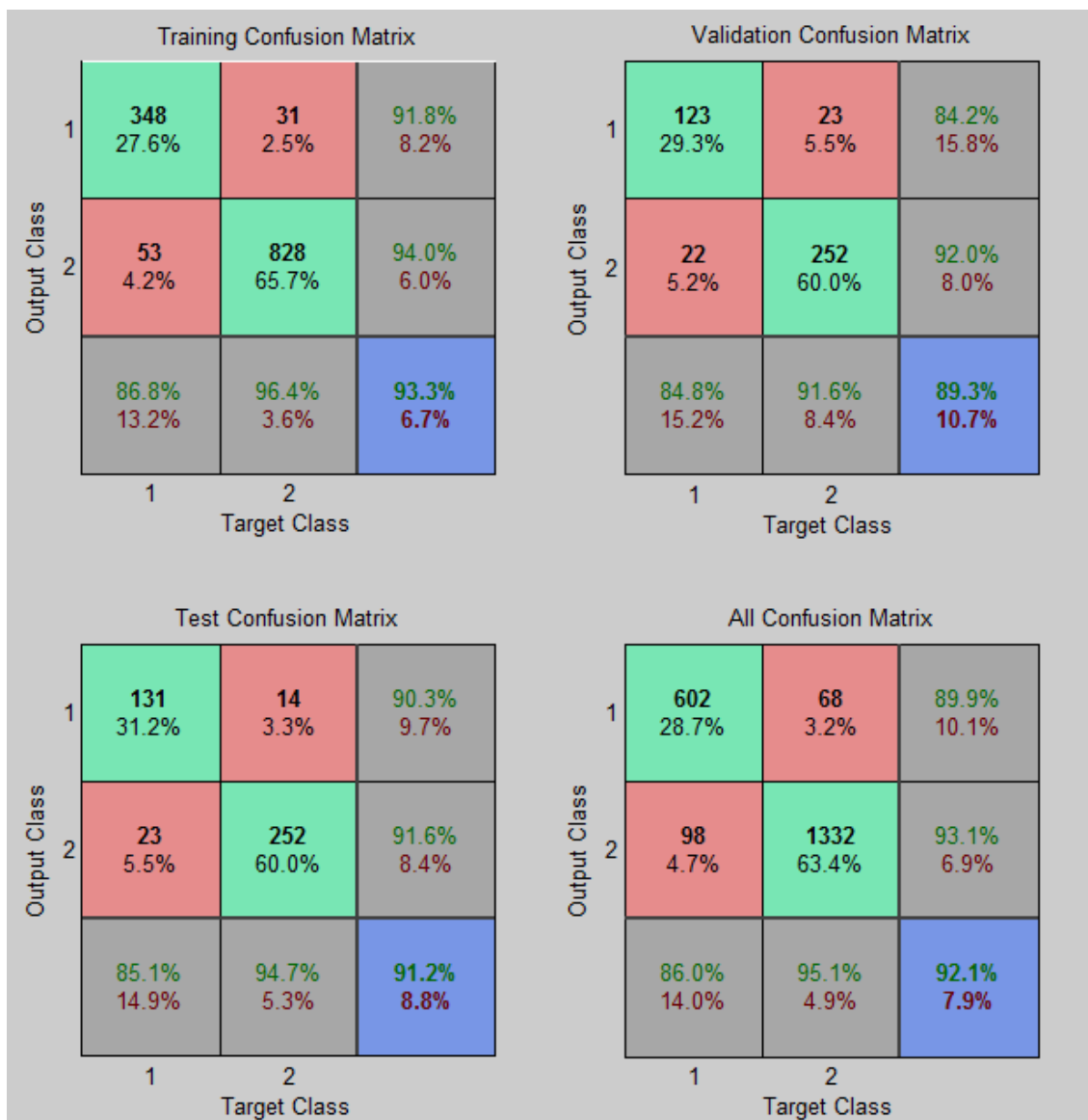


Figura 5.10: Matrices de confusión de la red neuronal para señales procesadas mediante el filtro Hilbert (caso 1).

Como se esperaba al ver la gráfica del rendimiento, el error en la validación es el mayor de todos. En esta ocasión, el porcentaje de acierto del entrenamiento es del 93,3%, el de la validación del 89,3% y el del test del 91,2%. Estos valores son menores que en los casos de redes creadas a partir de señales sin filtrar o sometidas al filtro Smooth. Al analizar la matriz de test se observa que la cantidad de señales incorrectamente clasificadas para el caso de eje con defecto es de 131 frente a 23

5. Redes neuronales artificiales

acertadas, mientras que la cantidad de aciertos para señales de eje sano es de 252 frente a 14 fallos. Cabe destacar que el porcentaje de fallos para las señales de eje con defecto (14,9%) es mucho mayor que el de señales de eje sano (94,7%).

El filtrado mediante la Transformada de Hilbert ha empeorado los resultados frente al filtro Smooth y las señales sin filtrar. La causa más probable es que este filtro, al cambiar más las señales, provoca una detección más complicada de los patrones entre señales parecidas. Además, a la vista de los resultados, queda claro que el filtro Hilbert es mejor para señales de eje sano que para señales de eje con defecto.

5.3.4 Filtrado Wavelet

Se va a crear una nueva red neuronal, esta vez con las señales filtradas mediante la Transformada Wavelet que se indicó anteriormente. Al entrenar la red se obtiene la toolbox mostrada en la figura 5.11.

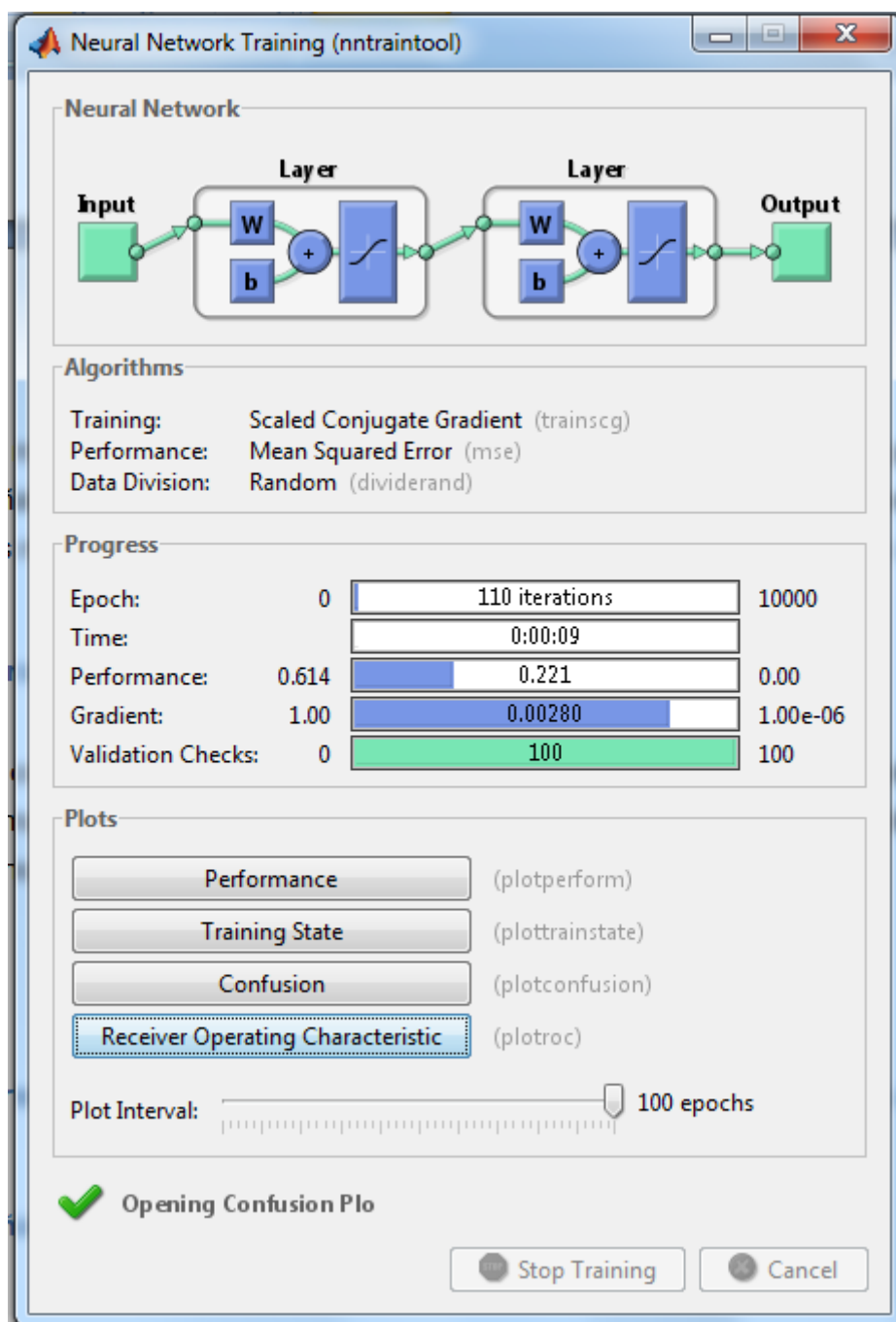


Figura 5.11: Toolbox de la red neuronal para señales procesadas con el filtro Wavelet (caso 1).

Este entrenamiento ha sido el más corto de todos, con sólo 110 iteraciones y 9 segundos. El rendimiento es el peor de todos, con un error de 0.221 y el gradiente tiene un valor intermedio: 0.00280.

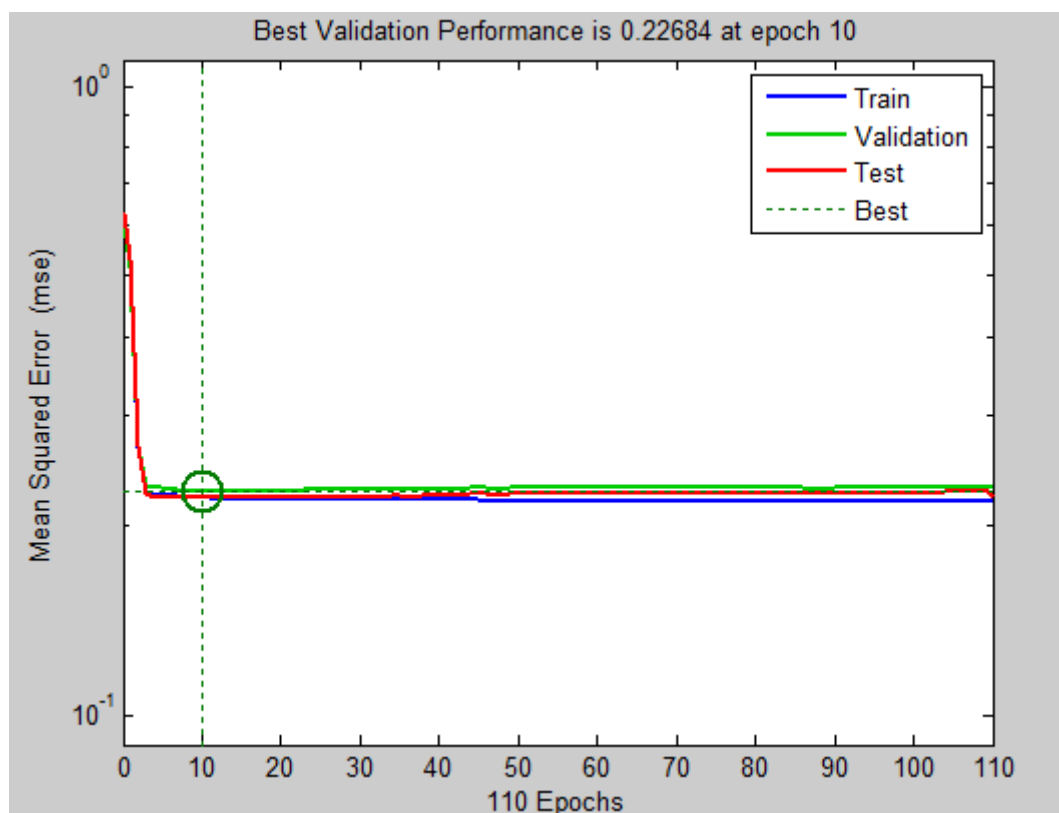


Figura 5.12: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Wavelet (caso 1).

En la gráfica del rendimiento (figura 5.12) se puede apreciar que los errores caen en picado durante las primeras tres o cuatro iteraciones y a partir de ahí se estabilizan, con un valor mínimo del error de validación de 0.22684 en la décima iteración. En esta ocasión los tres tipos de error van prácticamente a la par, con diferencias muy pequeñas entre ellos. Se incluye la gráfica del error de validación (figura 5.13) en la que se puede ver con claridad que sólo hay 10 iteraciones en las que el error de validación disminuye y a partir de ahí vienen las 100 iteraciones seguidas en las que no vuelve a alcanzar ese valor mínimo.

5. Redes neuronales artificiales

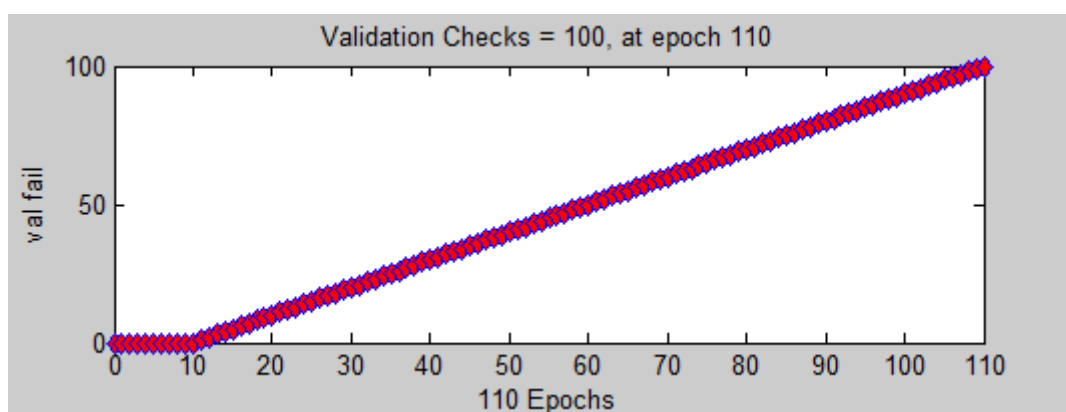


Figura 5.13: Gráfico del Estado de Entrenamiento de la red neuronal para señales procesadas con el filtro Wavelet (caso 1).

Se representan en la figura 5.14 las matrices de confusión, que son decisivas para ver el resultado de la red neuronal.

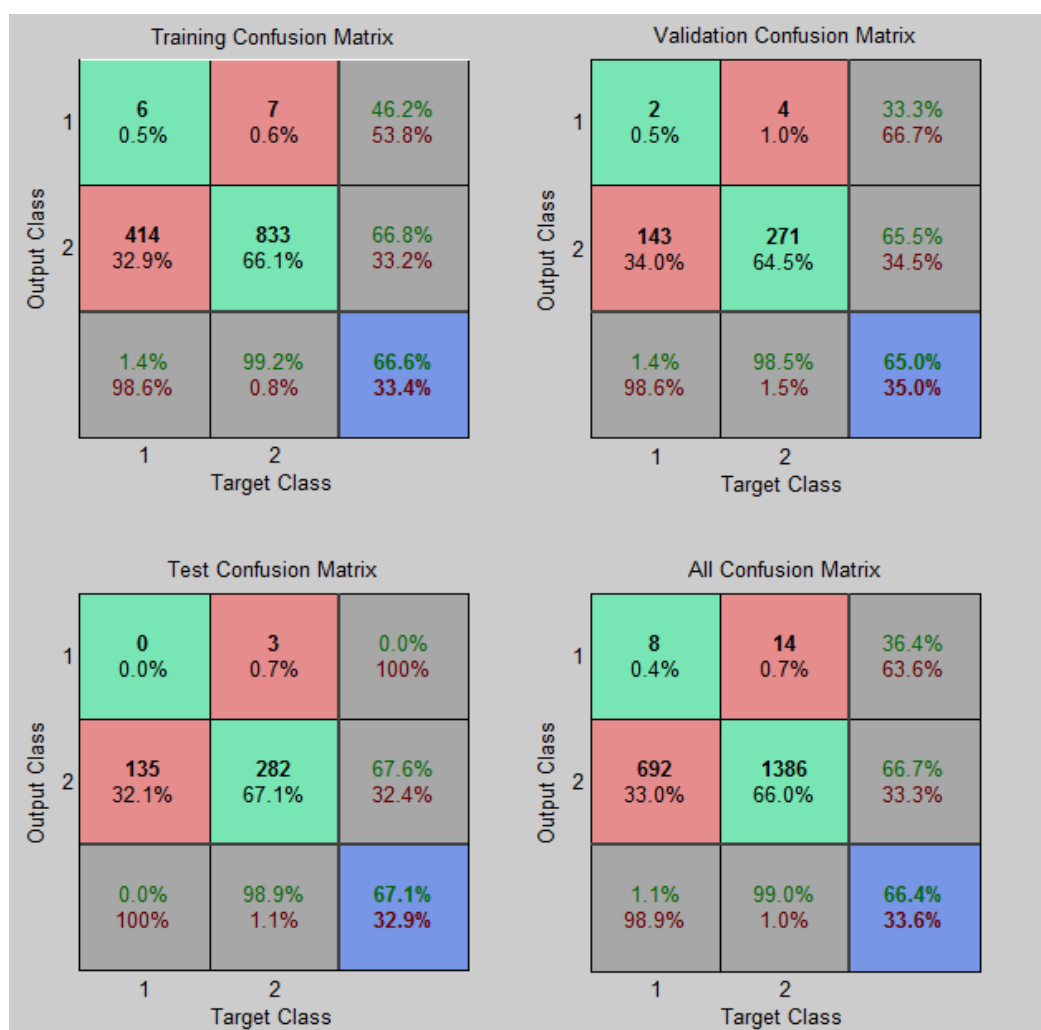


Figura 5.14: Matrices de confusión de la red neuronal para señales procesadas mediante el filtro Smooth (caso 1).

5. Redes neuronales artificiales

Se puede observar que la tasa de acierto es muy baja comparada con el resto de redes. Tan sólo alcanza valores del 66,6% para el entrenamiento, 65% para la validación y 67,1% para el test. Sobre todo, cabe destacar que casi todas las señales han sido clasificadas como eje sano. Al interpretar las matrices por filas, la primera corresponde a las señales clasificadas por la red como eje defectuoso y la segunda a las señales clasificadas como eje sano. Por columnas, la primera es la de señales que se ha indicado a la red que pertenecen a eje con defecto y la segunda de señales que se ha indicado que pertenecen a eje sano. Por este motivo, si los datos pertenecen a la primera columna y segunda fila se consideran fallos y en cambio los de la segunda columna y segunda fila son aciertos.

En la matriz de test solamente hay 3 señales clasificadas como eje con defecto, de las 417 estudiadas, y encima esas 3 pertenecen a eje sano. El porcentaje de acierto es muy cercano al 66% por el mero hecho de que se dispone del doble de señales de eje sano que de eje con defecto. Por tanto, si se clasifican todas como eje sano, el resultado es exactamente que el 66% de las señales se han diagnosticado correctamente.

La causa más probable de este fenómeno es que el filtrado de la señal mediante Transformada Wavelet elimine la información del ruido, considerándola ruido aleatorio. De esta forma, la red neuronal clasifica todas las señales como pertenecientes a eje sano.

5.4 Entrenamiento con unas carpetas y test con otras distintas

En esta segunda parte del estudio de redes neuronales, se utilizan unas carpetas para el entrenamiento de las redes y otras con datos de días distintos para el test. De este modo se consigue que las señales de entrenamiento y test tengan niveles totalmente distintos de ruido, poniendo a prueba verdaderamente la validez de los filtros. Por este motivo, se esperan tasas de acierto mucho menores que en el caso anterior.

Se empleará un 57% de las carpetas para el entrenamiento y creación de la red y el 43% restante para el test. De las carpetas de entrenamiento se utilizará un 98% de los datos para entrenamiento y sólo un 1% para validación y el 1% restante para test. Es importante que haya algún dato para validación y test en el transcurso del entrenamiento, pero se indica una vez más que el test que se tendrá en cuenta es el realizado con las otras carpetas.

5.4.1 Señales originales sin filtrar

La primera red neuronal se creará y testeará con las señales sin filtrar, para tener una referencia en la tasa de acierto que pueda ayudar a determinar qué filtros mejoran o empeoran los resultados.

De nuevo, se utiliza la toolbox de la red (figura 5.15) para determinar los datos del entrenamiento de la misma.

5. Redes neuronales artificiales

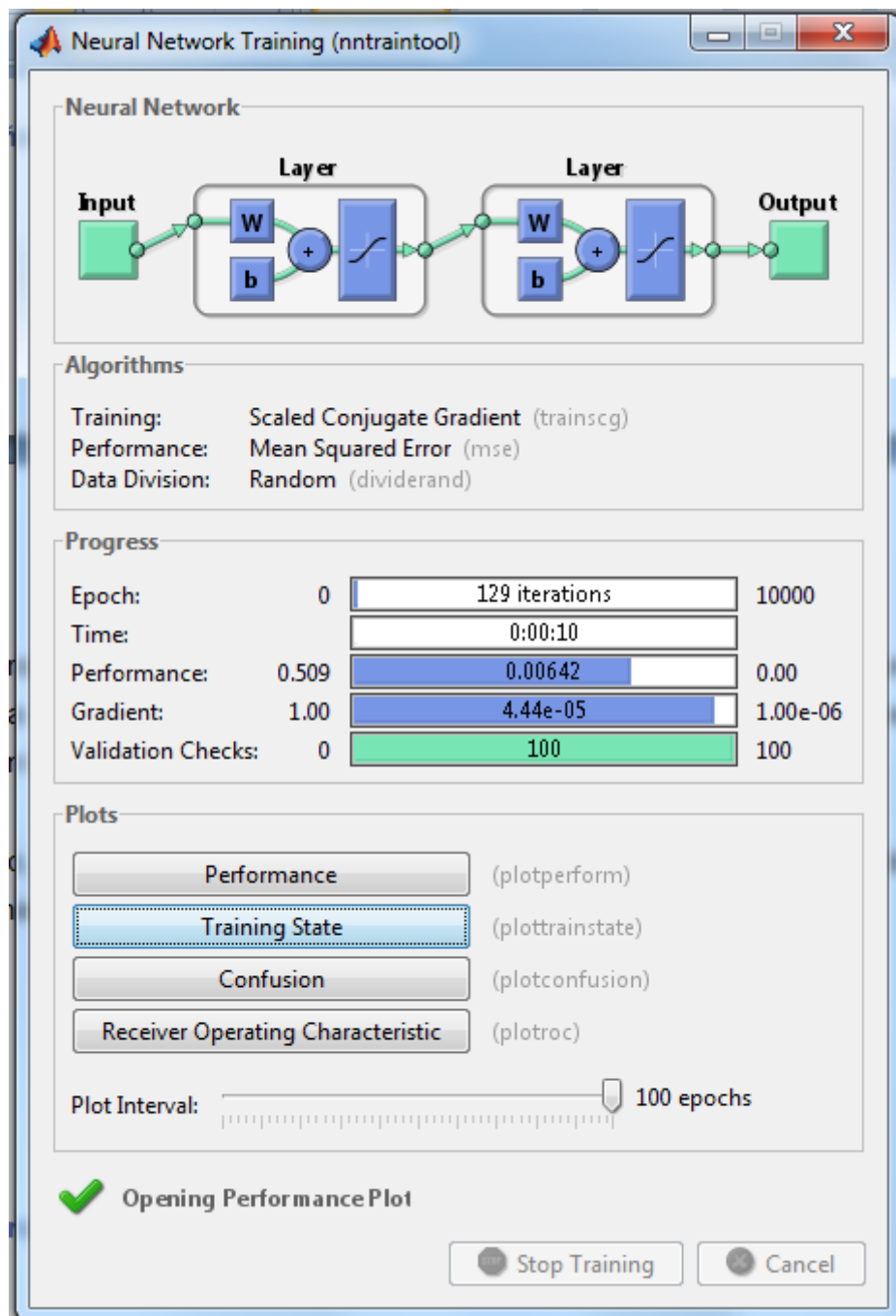


Figura 5.15: Toolbox de la red neuronal para señales sin filtrar (caso 2).

El entrenamiento se detuvo tras sólo 129 iteraciones y 10 segundos. Los entrenamientos van a ser bastante más rápidos en esta parte del estudio porque se utilizan menos datos. El rendimiento tiene un error bastante bajo, con un valor de 0.00642; mientras que el gradiente es el menor de los vistos hasta ahora, con un valor de $4.44 \cdot 10^{-5}$.

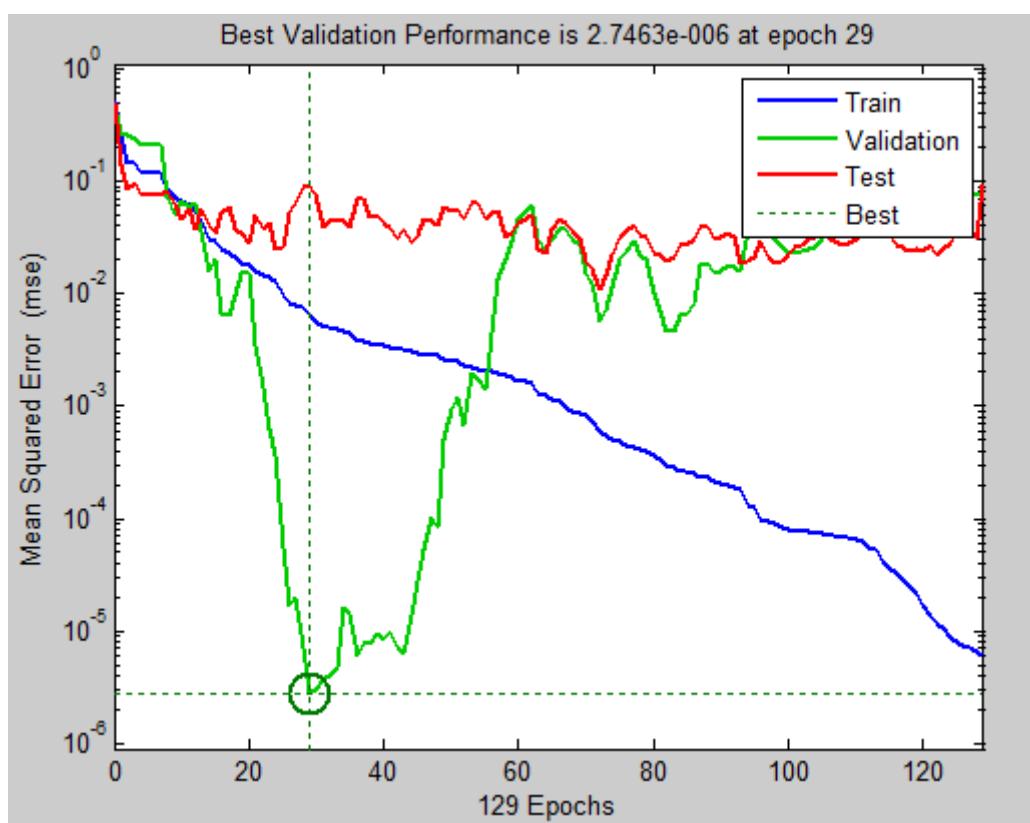


Figura 5.16: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales sin filtrar (caso 2).

La gráfica del rendimiento (figura 5.16) muestra un comportamiento errático en los errores de la validación y el test, lo cual se debe a que sólo un 1% de las señales de las carpetas utilizadas para el entrenamiento se usan para validar y otro 1% para testear, haciendo que en cada iteración su error sea aleatorio. El error del entrenamiento, en cambio, disminuye progresivamente a medida que pasan las iteraciones. Dado el comportamiento del error de validación, el número de iteraciones de entrenamiento será también aleatorio.

Las matrices de confusión utilizadas anteriormente para la tasa de acierto de la red neuronal sólo tenían en cuenta los datos utilizados para crear la red, por lo que no serán útiles para esta parte del estudio. En este caso, se ha creado un algoritmo que produce una gráfica que nos muestra la clasificación de las señales por parte de la red neuronal frente a su verdadero estado. Esta gráfica se representa en la figura 5.17.

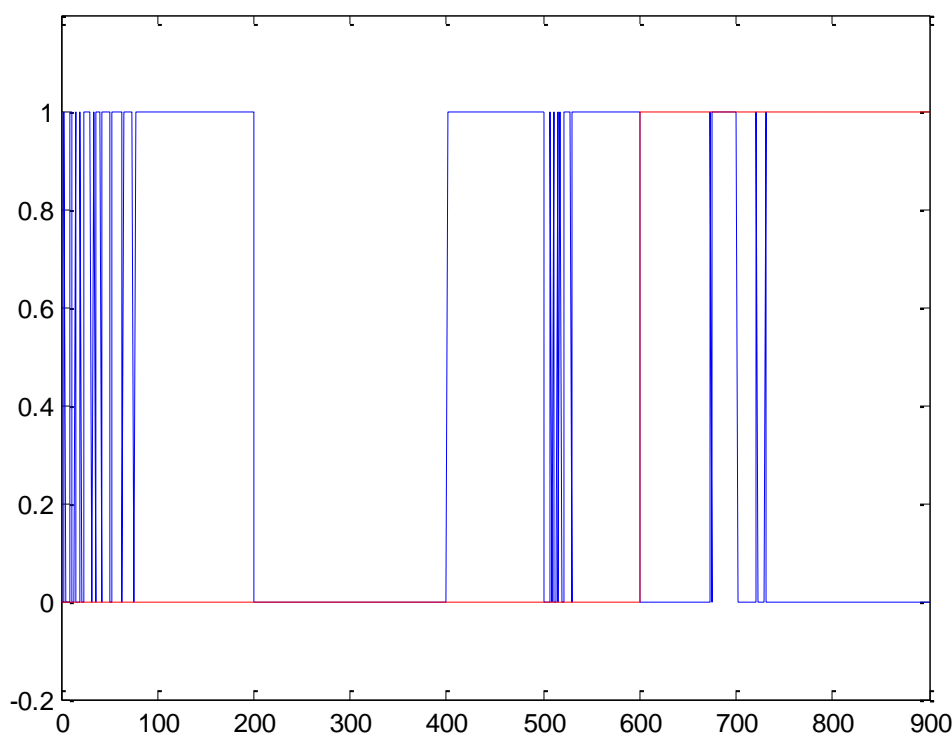


Figura 5.17: Gráfica de acierto para señales sin filtrar (caso 2).

El eje X de la gráfica muestra las señales y el eje Y tendrá un valor de 0 en el caso de corresponder a eje sano y de 1 en caso de corresponder a eje con defecto. Las líneas rojas corresponden a los diagnósticos reales de las señales y las azules a la clasificación propuesta por la red neuronal. Se puede apreciar que se han utilizado 900 señales para el test de la red, las primeras 600 correspondientes a eje sano y las 300 restantes a eje con defecto.

De las señales de eje sano más de la mitad son clasificadas por la red como eje defectuoso y de las correspondientes a eje con defecto, casi todas son clasificadas como eje sano. La tasa de acierto es tan sólo del 30,1111%, un valor bajísimo. Se ve que al presentar las señales niveles muy diferentes de ruido, la red no es capaz de diferenciar los patrones de las señales de eje sano y los de las señales de eje con defecto.

Por tanto, es necesario el uso de filtros para eliminar parte de los ruidos aleatorios y facilitar a la red el reconocimiento de los patrones. De esta forma, se espera mejorar el porcentaje de aciertos.

5.4.2 Filtrado Smooth

El primer filtro que se va a poner a prueba es el filtro Smooth. Se crea la red neuronal con las señales filtradas y se obtienen los parámetros del entrenamiento de la toolbox correspondiente, mostrada en la figura 5.18.

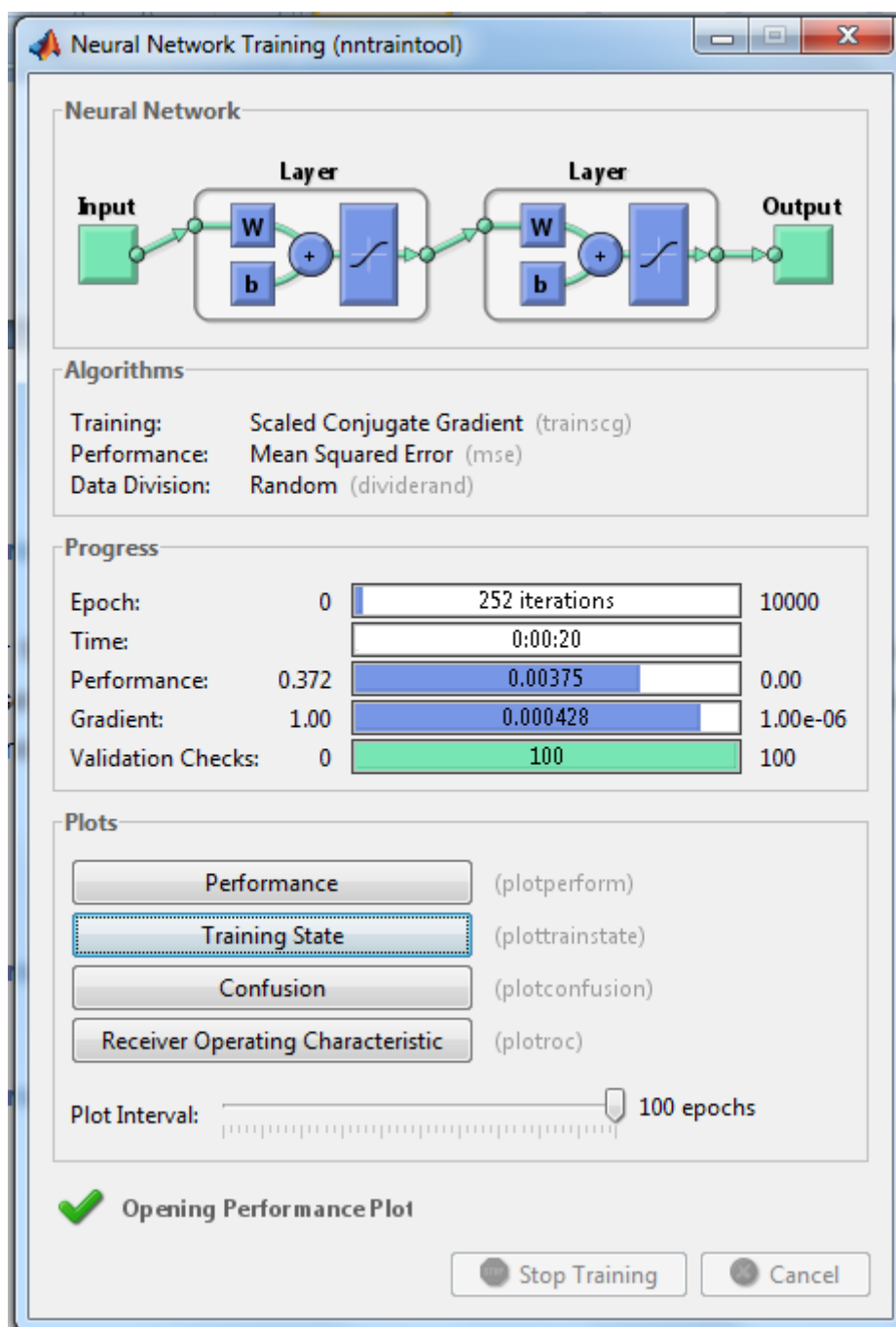


Figura 5.18: Toolbox de la red neuronal para señales procesadas con el filtro Smooth (caso 2).

5. Redes neuronales artificiales

El entrenamiento se ha completado en 252 iteraciones y 20 segundos. El rendimiento ha alcanzado un error de 0,00375 y el gradiente 0,000428. Todos los valores son más o menos intermedios comparados con los vistos hasta ahora.

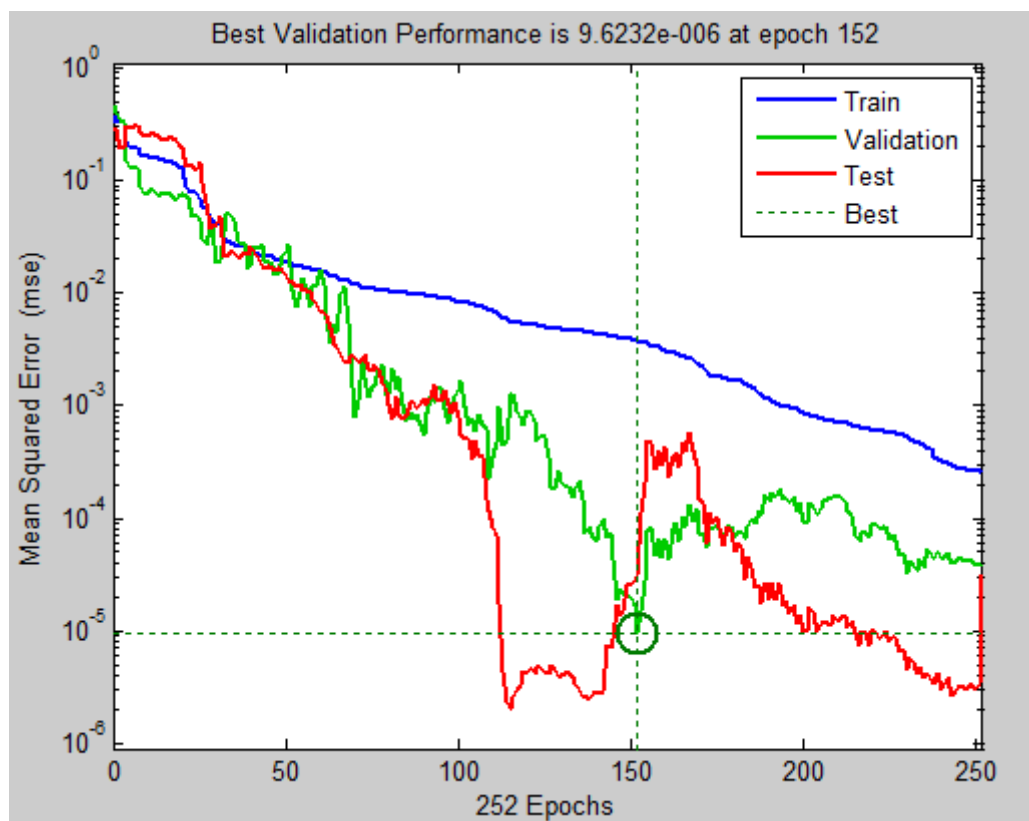


Figura 5.19: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Smooth (caso 2).

El error observado en la figura 5.19 tiene una evolución más progresiva que en el caso de señales sin filtrar. Los valores para la validación y el test siguen siendo un poco erráticos, mientras que el decrecimiento del error de entrenamiento es bastante uniforme. El error de validación alcanza un valor mínimo de $9,6232 \cdot 10^{-6}$, el cual es realmente bajo, en la iteración 152

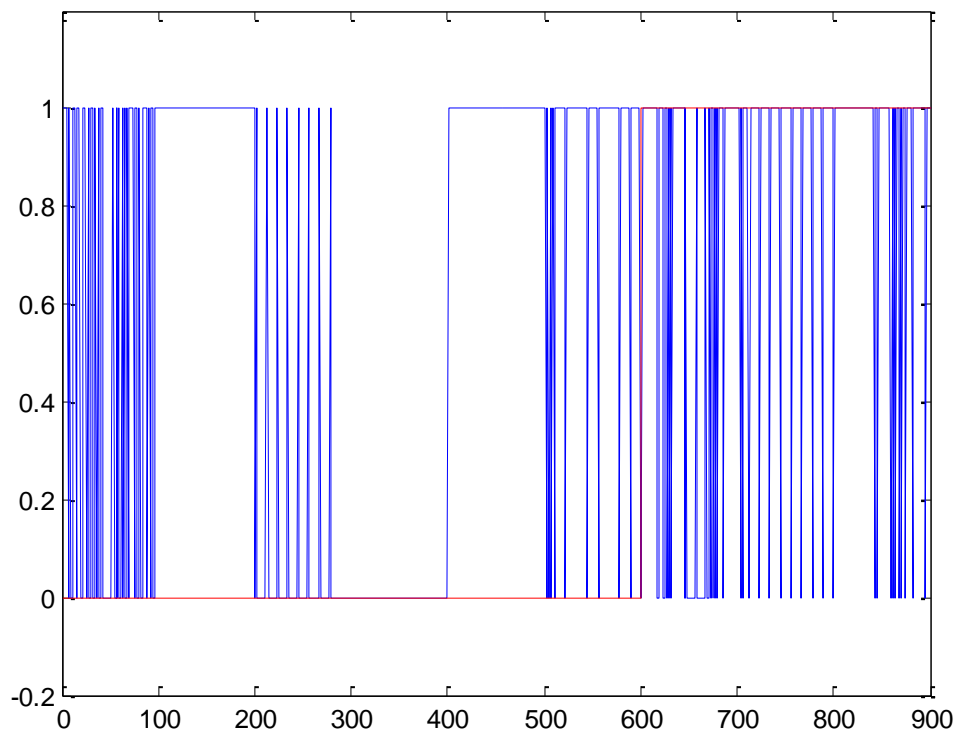


Figura 5.20: Gráfica de acierto para señales procesadas con el filtro Smooth (caso 2).

En el gráfico de la tasa de acierto (figura 5.20) se observa que la cantidad de aciertos y errores de la red es bastante parecida. El valor exacto de la tasa de acierto es del 54,8889%, lo cual supone una gran mejora con respecto a las señales sin filtrar, que sólo alcanzaron el 30,1111%.

Aunque la cantidad de aciertos no es buena, se ha logrado una tasa de acierto de casi el doble de la alcanzada antes de filtrar las señales. Esto indica que la eliminación de ruido conseguida mediante el filtro Smooth sí que ha supuesto una gran ayuda para el reconocimiento de patrones de la red neuronal.

5.4.3 Filtrado Hilbert

Se va a estudiar la validez del filtrado mediante la Transformada de Hilbert para eliminar el ruido de las señales y facilitar el reconocimiento de patrones de la red neuronal. La figura 5.21 muestra la toolbox del entrenamiento de la red.

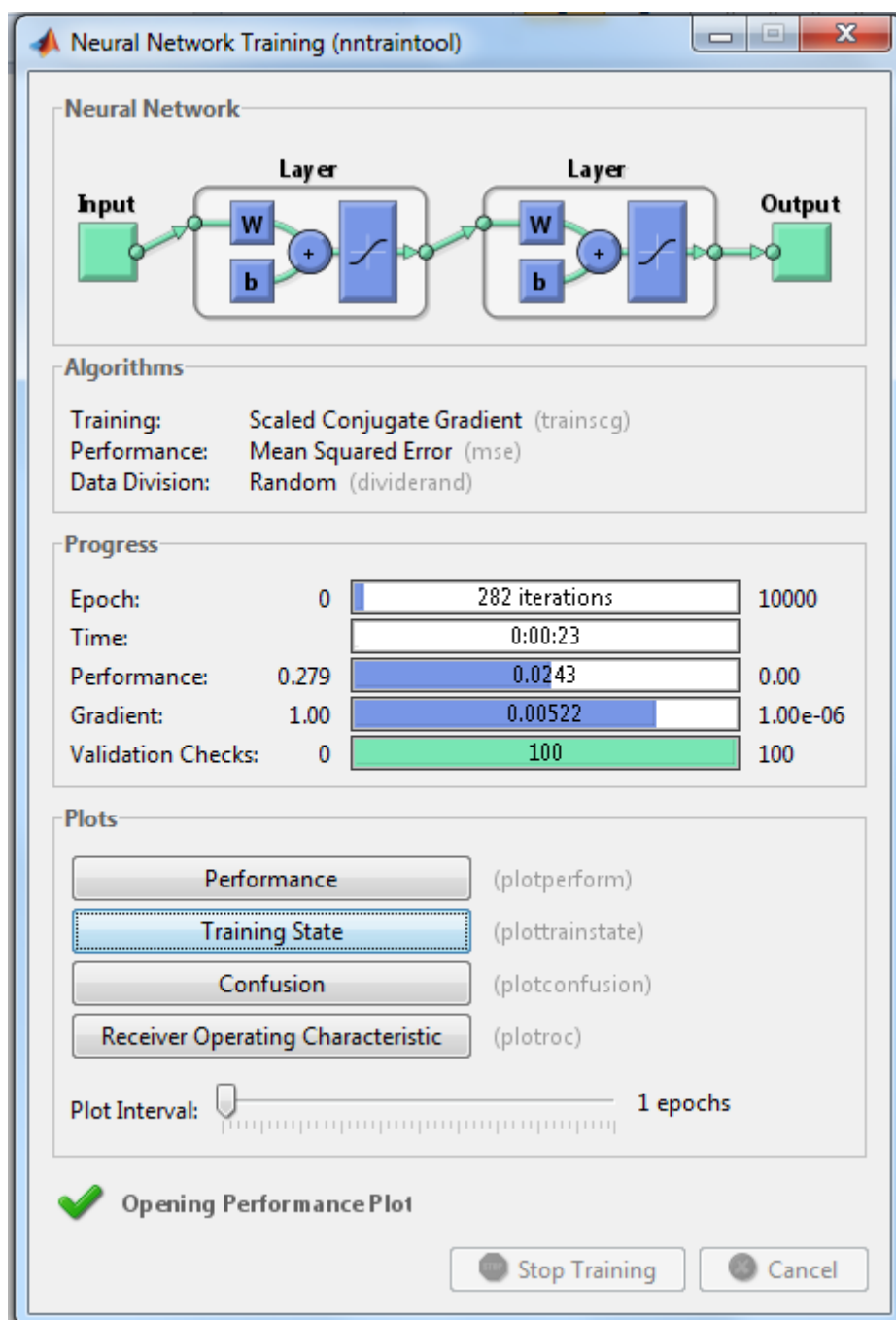


Figura 5.21: Toolbox de la red neuronal para señales procesadas con el filtro Hilbert (caso 2).

5. Redes neuronales artificiales

La red se ha entrenado en 23 segundos y 282 iteraciones, tiene un rendimiento de 0,0243 y un gradiente de 0,00522. Estos últimos valores son bastante altos en comparación con los de las señales originales o filtradas con Smooth.

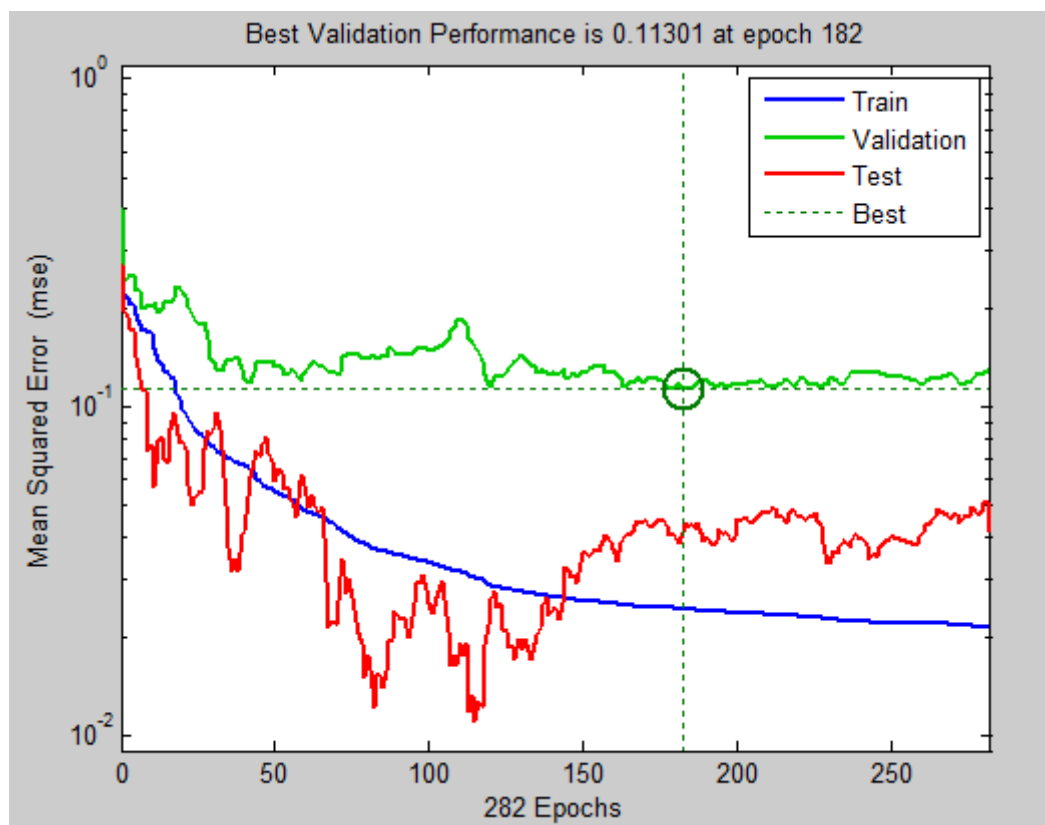


Figura 5.22: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Hilbert (caso 2).

La figura 5.22 muestra el error del rendimiento y en ella se aprecia un decrecimiento hiperbólico del error del entrenamiento, con un valor que ya presenta poca variación en las últimas iteraciones. El error de validación mantiene un valor similar en casi toda la gráfica, con ligeras subidas y bajadas, y un valor mínimo de 0,11301 en la iteración 182.

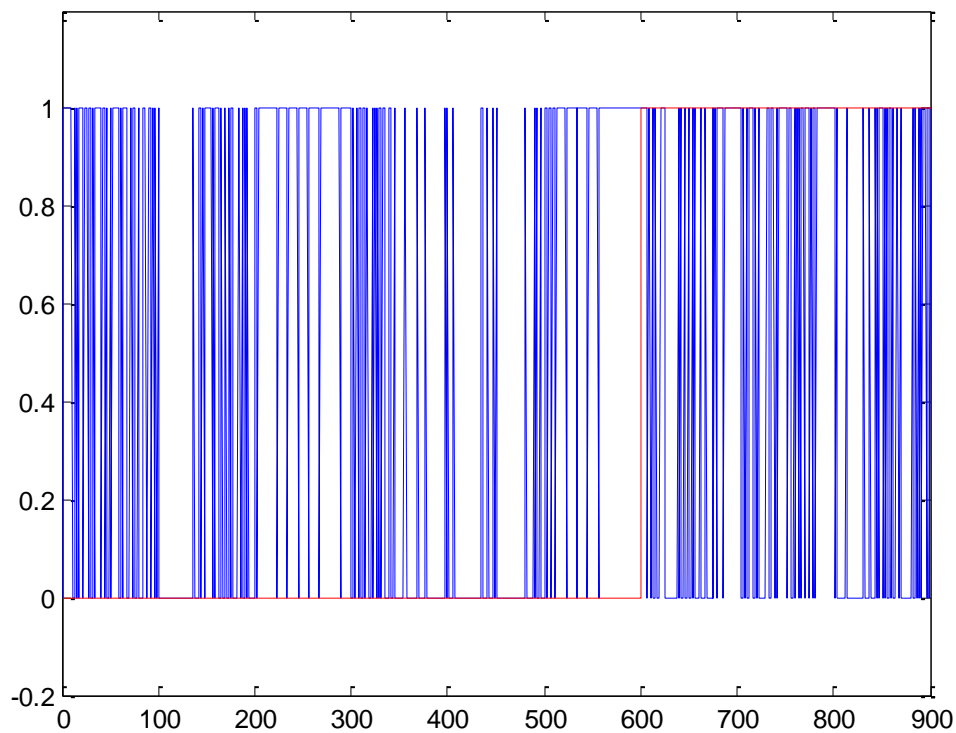


Figura 5.23: Gráfica de acierto para señales procesadas con el filtro Hilbert (caso 2).

El gráfico de la tasa de acierto (figura 5.23) es bastante poco determinante, ya que no se distinguen patrones claros en los diagnósticos de la red. El porcentaje de acierto es del 47,8889%, mejorando con creces los resultados obtenidos sin filtrar la señal (30,1111%). Aún así, este filtrado presenta menos aciertos que el del filtro Smooth (54,8889%), motivo por el cual se cree que elimina menos ruido aleatorio.

5.4.4 Filtrado Wavelet

Por, último se va a crear y ensayar la red neuronal con las señales filtradas mediante Transformada Wavelet. Los primeros resultados se obtienen de la figura 5.24, que muestra la toolbox del entrenamiento.

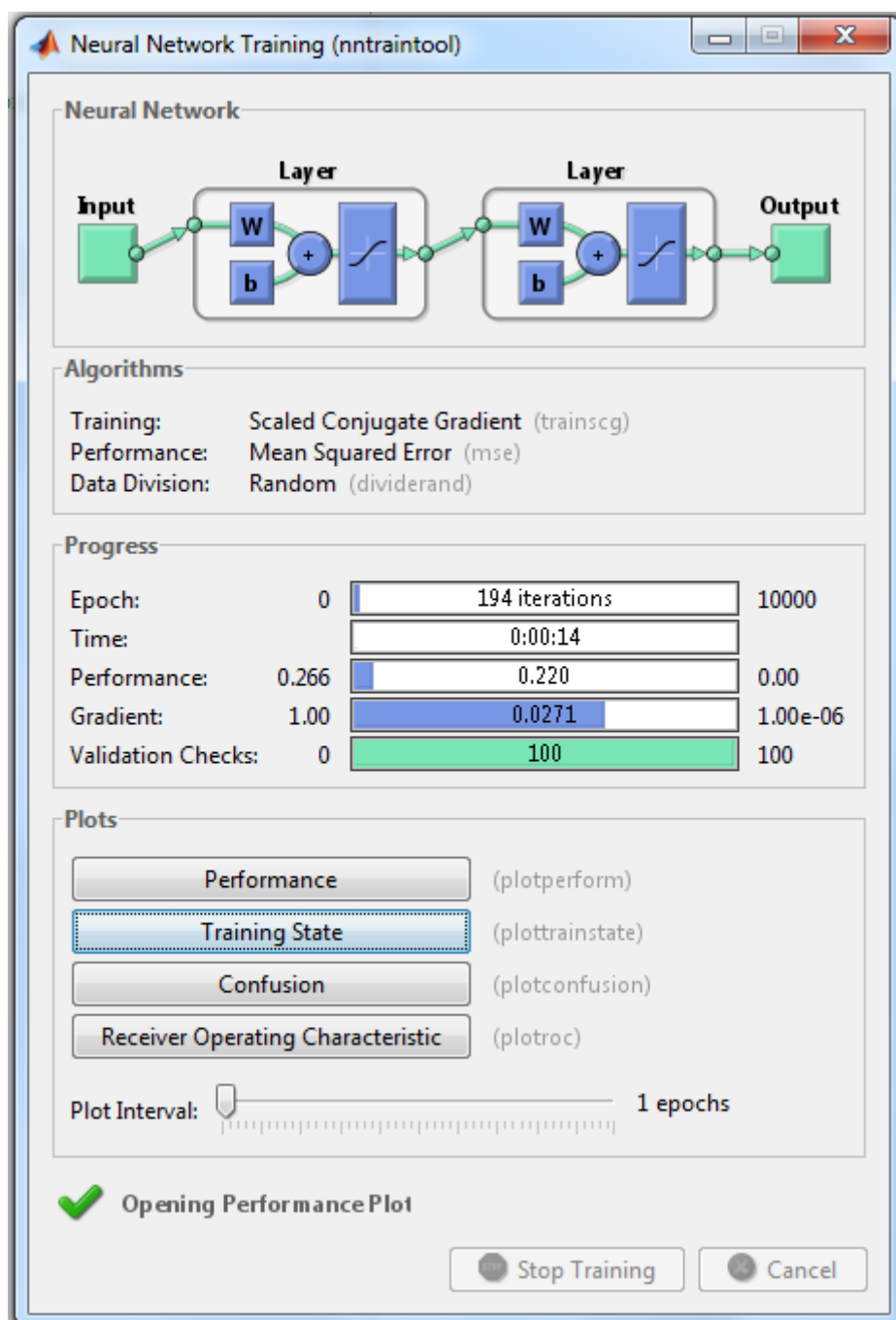


Figura 5.24: Toolbox de la red neuronal para señales procesadas con el filtro Wavelet (caso 2).

5. Redes neuronales artificiales

El entrenamiento de la red se ha completado tras 194 iteraciones y 14 segundos. El rendimiento y el gradiente tienen valores muy altos: 0,220 y 0,0271 respectivamente, lo cual indica que el entrenamiento no va a ser bueno.

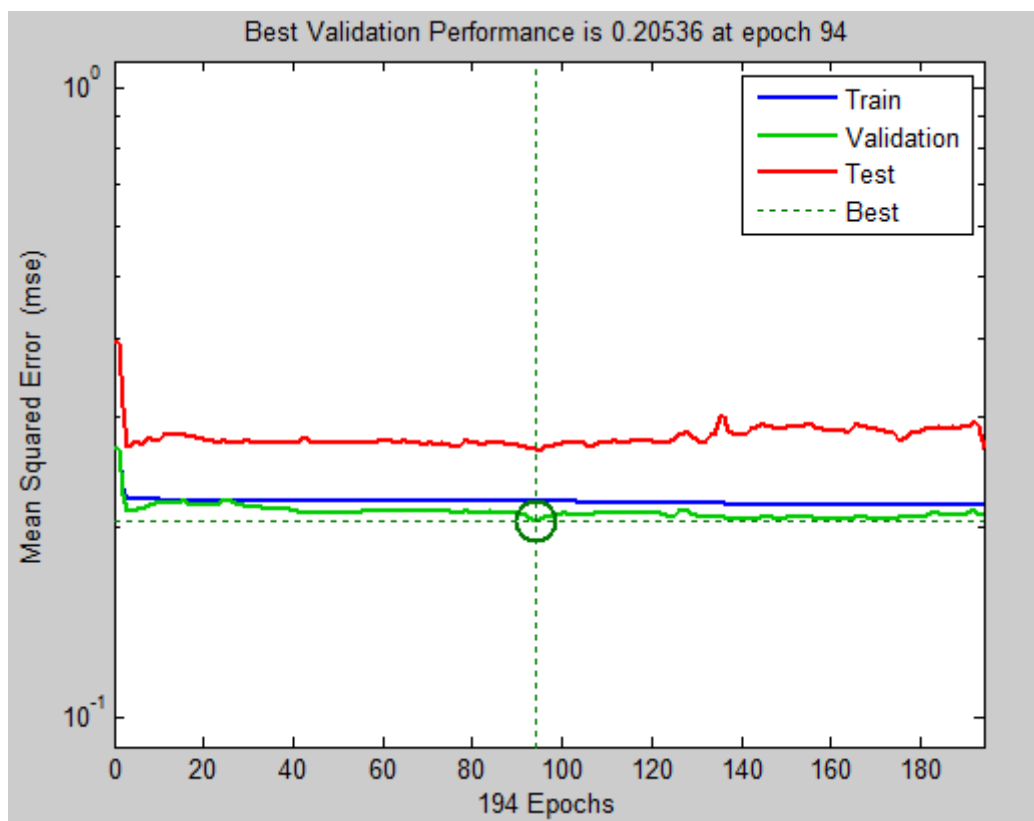


Figura 5.25: Gráfica del Rendimiento durante el entrenamiento de la red neuronal para señales procesadas con el filtro Wavelet (caso 2).

La gráfica del error observada en la figura 5.25 muestra un comportamiento muy parecido al de la otra red creada con señales filtradas por Wavelet (figura 5.12), con un decrecimiento brusco en las primeras iteraciones y unos valores muy estables posteriormente. Los valores en validación y test presentan pequeñas subidas y bajadas pero el entrenamiento no varía casi en gran parte del gráfico. El menor error de validación es de 0,20536 y se produce en la iteración 94.

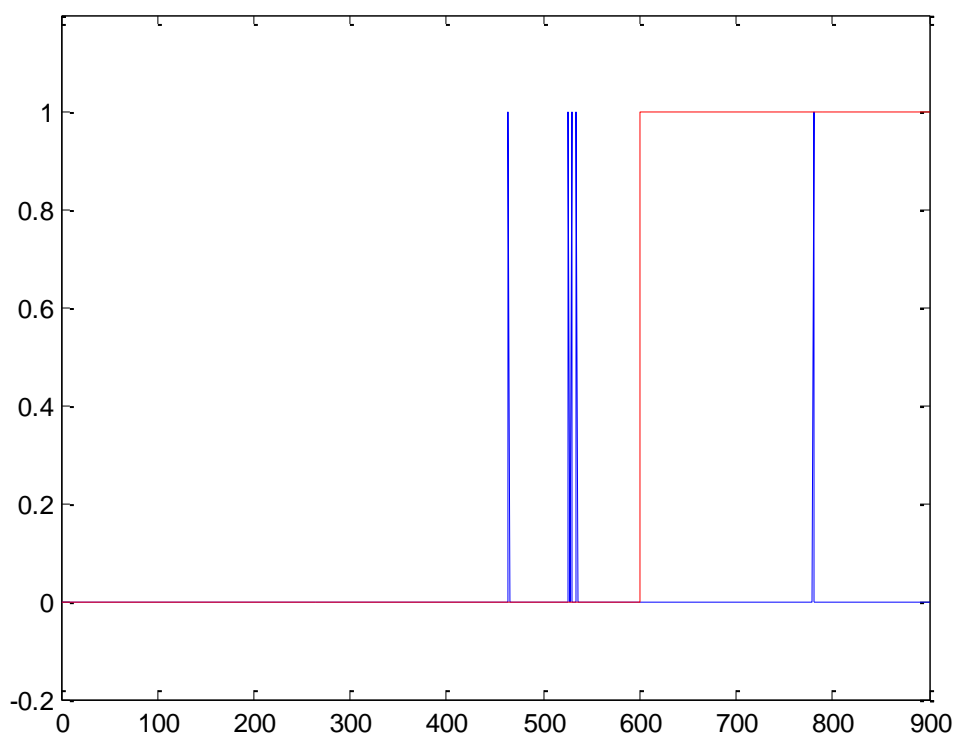


Figura 5.26: Gráfica de acierto para señales procesadas con el filtro Wavelet (caso 2).

El gráfico de la tasa de acierto (figura 5.26) muestra que la red neuronal clasifica casi todas las señales como pertenecientes a eje sano, al igual que pasó en la parte anterior del estudio de redes con el filtro Wavelet. Solamente se clasifican como eje con defecto unas pocas señales singulares y la mayoría pertenecen a eje sano. El 66,6666% de las señales pertenecen a eje sano, de ahí que la tasa de acierto sea del 66,3333%.

Se confirma entonces que el filtrado mediante la Transformada Wavelet elimina la información del defecto en la señal. Se ha visto que sea cual sea el entrenamiento de la red, no consigue reconocer el patrón de las señales con defecto tras aplicarlas el filtro Wavelet y considera todas las señales como pertenecientes a eje sano.

6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

Como el propósito principal del proyecto es la comparación de los distintos filtros propuestos, se van a utilizar tablas comparativas para facilitar la comprensión de las conclusiones. En primer lugar se analizan los resultados obtenidos para redes neuronales entrenadas y testeadas con todas las carpetas de señales.

Tabla 6.1: Características de la red neuronal para cada tipo de filtrado (caso 1).

	Sin filtrar	Filtro Smooth	Filtro Hilbert	Filtro Wavelet
Tasa de acierto	97,10%	96,40%	91,20%	67,10%
Margen	96%-100%	94%-99%	90%-93%	65%-68%
Iteraciones	472	788	276	110
Tiempo	40	1:05	23	9
Rendimiento	0,00327	0,00576	0,0623	0,221
Gradiente	0,000742	0,00355	0,00565	0,0028

La tabla 6.1 muestra una comparación de las tasas de acierto y los valores de entrenamiento de las redes neuronales dependiendo del filtro utilizado para procesar las señales. Se facilitan también los márgenes entre los que ha variado la tasa de acierto en los distintos entrenamientos, aunque se haya escogido el más repetido para el análisis. En negrita se han incluido los valores más favorables. Las señales sin filtrar muestran el mejor comportamiento al entrenar la red con señales parecidas a las usadas para testearla. La tasa de acierto supera a todas las demás, y tanto el error del rendimiento como el gradiente son claramente menores.

6. Conclusiones

Al evaluar cada filtro por separado se sacan las siguientes conclusiones:

- **Sin filtrar:** el porcentaje de acierto es altísimo (97,1%), por tanto, entre señales parecidas los resultados son muy buenos. Esto indica que con un entrenamiento idóneo se conseguiría una gran eficacia.
- **Smooth:** el porcentaje de acierto es muy alto (96,4%), por lo que entre señales con el mismo nivel de ruido se alcanzan muy buenos resultados. Con un entrenamiento idóneo se conseguiría una eficacia cercana a la de señales sin filtrar.
- **Hilbert:** el porcentaje de acierto es alto (91,2%), lo que indica que entre señales parecidas acierta mucho. Aunque se hiciera un entrenamiento muy bueno, los resultados no serían muy destacables.
- **Wavelet:** el porcentaje de acierto es bajo (67,10%). Siempre diagnostica el eje como sano, por lo que la red neuronal no aporta ninguna información. Se cree que el filtro Wavelet elimina los datos de la señal que caracterizan el error. Por mucho que se entrene la red, los resultados son malos.

Por tanto, cuando las señales de entrenamiento y test tienen niveles parecidos de ruido, lo mejor es utilizar las señales sin filtrar, ya que se pueden perder detalles de la vibración que empeoren el reconocimiento de los patrones. El problema es que esto es muy difícil de conseguir, ya que las señales de test pueden presentar niveles muy distintos de ruido. Serían necesarias innumerables señales para el entrenamiento de la red neuronal.

A continuación se analizan los resultados obtenidos de las redes neuronales entrenadas con unas carpetas y testeadas con otras distintas. Este va a ser el modo determinante de decidir qué filtro es mejor para el diagnóstico de ejes, ya que se asemeja más al caso real, en el que las señales utilizadas en el entrenamiento del sistema son distintas de las que hay que diagnosticar.

6. Conclusiones

Tabla 6.2: Características de la red neuronal para cada tipo de filtrado (caso 2).

	Sin filtrar	Filtro Smooth	Filtro Hilbert	Filtro Wavelet
Tasa de acierto	30,11%	54,89%	47,89%	66,33%
Margen	29%-31%	51%-56%	46%-50%	65%-68%
Iteraciones	129	252	282	194
Tiempo	10	20	23	14
Rendimiento	0,00642	0,00375	0,0243	0,22
Gradiente	$4,44 \cdot 10^{-5}$	0,000428	0,00522	0,0271

La tabla 6.2 muestra las tasas de acierto y los parámetros de entrenamiento para cada tipo de filtro utilizado, además de los márgenes de variación de la tasa de acierto en los distintos entrenamientos de la red. Una vez más, se han marcado en negrita los resultados más favorables. El filtro Smooth alcanza el mayor porcentaje de clasificaciones correctas y el error más bajo en el rendimiento, mientras que el gradiente más bajo se alcanza con las señales sin filtrar. Aunque el filtro Wavelet alcanza una tasa de acierto mayor, no se considera válida, ahora se explicará la razón.

Las conclusiones sacadas al evaluar cada filtro por separado son:

- **Sin filtrar:** el porcentaje de acierto es el más bajo de todos (30,11%), lo cual se debe a que el ruido en la señal hace que sea muy difícil diferenciar los patrones característicos de las señales de eje sano y eje con defecto.
- **Smooth:** el porcentaje de acierto es el más alto de todos (54,89%), cerca del doble que el de las señales sin filtrar, lo cual mejora notablemente la calidad de la señal. El error en el rendimiento es el más bajo de todos.
- **Hilbert:** el porcentaje de acierto es cercano al más alto (47,89%), lo que significa que también se obtiene una reducción del ruido considerable. El error del rendimiento es demasiado alto.
- **Wavelet:** aunque a simple vista parece que el porcentaje de acierto es el más alto (66,33%), esto se debe únicamente a que el 66,66% de las señales pertenecen a eje sano y la red clasifica casi todas las señales en este estado. La tasa de acierto obtenida no es válida, ya que sólo depende del número de ejes sanos y en este caso el porcentaje sólo nos sirve para ver que la red neuronal no funciona de manera adecuada tras filtrar con Wavelet.

6. Conclusiones

A la vista de los datos, el procesamiento con el que se consiguen mejores resultados es el filtro Smooth. Gracias a él, se elimina una buena parte del ruido aleatorio, aunque los resultados siguen sin ser destacables.

6.2 Trabajos futuros

Los porcentajes de acierto obtenidos siguen sin ser cercanos a los necesarios para confiar en esta técnica para el mantenimiento de ejes en ferrocarriles. Será necesaria una investigación más profunda para encontrar filtros más adecuados o redes neuronales que funcionen mejor.

Se propone utilizar distintos tipos de filtrado o combinarlos entre sí (si son compatibles) para ver si se consigue una mejora en los resultados. Por ejemplo realizando un filtrado paso bajo y posteriormente la Transformada de Hilbert, o incluso ésta última precedida de un filtrado Smooth.

También se obtendrían mejores resultados con un aporte mayor de señales de vibración al estudio. Al disponer de más señales, se conseguirían señales con niveles de ruido más variado y se ha observado que cuando los niveles de ruido son parecidos, la tasa de acierto es realmente elevada.

7. REFERENCIAS

- [1] White G. "Introducción al análisis de vibraciones". Woburn, Estados Unidos. Azima DLI. 2010.
- [2] Crespo A, Moreu P, Sánchez A. "Ingeniería de Mantenimiento. Técnicas y métodos de aplicación a la fase operativa de los equipos." Ed. AENOR, Madrid. 412 págs. ISBN 84-8143-390-X. 2004.
- [3] Calvo Rolle, JL. "Sistema basado en conocimiento para análisis de vibraciones en mantenimiento predictivo". DYNA Ingeniería e Industria (vol. 83, nº 8, pág. 484-492). Noviembre 2008.
- [4] Braun S. "Discover signal processing. An interactive guide for engineers". Israel Institute of Techonology, Israel. England, Wiley, 2008.
- [5] Matlab help, Smooth filter. Matlab version 7.10.0.499 (R2010a). The Mathworks, Inc. February 5, 2010.
- [6] "Caracterización de defectos localizados en sistemas mecánicos rotativos mediante análisis de vibraciones". Tesis doctoral. Higinio Rubio Alonso. Universidad Carlos III de Madrid, Departamento de Ingeniería Mecánica. 2003.
- [7] Kraniuskas P. " Transforms in signals and systems". Suffolk, Great Britain. Addison-Wesley, 1992. 558 p.
- [8] Misiti M, Misiti Y, Oppenheim G, Poggi JM. "*Wavelet Toolbox for use with Matlab®*". The Math Works Inc. EEUU. 2004.
- [9] Sweldens W, Schrder P. "Building your own wavelets at home", Wavelets in Computer Graphics, pp. 15--87, ACM SIGGRAPH Course notes, 1996.
- [10] C. Sydney Burrus, Armes A. Gopinath, Haidao Guo. "Introduction to wavelets and wavelet transforms". Prentice Hall. 1998.
- [11] G. C. Yen, K. C. Lin. "Wavelet packets feature extraction for vibration monitoring". IEEE transactions on industrial electronics. 2000, vol. 47, p. 650-667.

- [12] Norden E.Huang. "Introduction to the Hilbert-Huang transform and its related mathematical problems". Documento interno, Goddard Institute for Data Analysis, Code 614.2, NASA.
- [13] Sorsa T, Koivo H.N. "Application of Artificial Neural Networks in Process Fault Diagnosis". Automatica, 1993, Vol. 29, p. 843-849.
- [14] Isasi P, Galván I. "Redes de Neuronas Artificiales. Un enfoque práctico." Pearson Prentice Hall. Madrid, España. 2004. ISBN: 84-205-4025-0.
- [15] Godoy M, Meireles M, Almeida P. "A comprehensive review for industrial applicability of artificial neural networks". IEEE transactions on industrial electronics. 2003, vol. 50, p. 585-601.
- [16] Mcculloch W, Pitts W. "A logical calculus of the ideas immanent in nervous activity". Bulletin of Mathematical Biophysics, 1943, Vol. 9, p. 127-147.
- [17] Lara O. J. "Nuevas metodologías no invasivas de diagnosis de defectos incipientes en rodamientos de bola". Tesis Doctoral. Universidad Carlos III de Madrid, Departamento de Ingeniería Mecánica, (2007).
- [18] Martinez I. "Detección de fallos en rodamientos mediante análisis de energía y lógica difusa". Proyecto fin de carrera. Universidad Carlos III de Madrid, Departamento de Ingeniería Mecánica, (2007).
- [19] Romero J. "Herramienta para la adquisición, procesamiento y monitorización de señales. Detección de fallos en ejes y rodamientos mecánicos." PFC. Ingeniería Industrial, especialidad Automática y Electrónica Industrial. Universidad Carlos III de Madrid. 2011. 195p.
- [20] Steven W. Smith. "The Scientist and Engineer's Guide to Digital Signal Processing". Ph.D.

8. ANEXOS

Anexo I: Programa de Matlab® para filtrar y sacar las gráficas de una señal de vibración

Para la primera parte del estudio se necesitaban las gráficas sólo de las señales más representativas. Por tanto, cada señal se cargaba por separado y cada vez que se lanzaba el programa se obtenían las gráficas filtradas y sin filtrar en el dominio del tiempo y en el de la frecuencia. Además, se guarda el archivo .mat de cada señal, por si fuera necesario a lo largo del proyecto.

Se ha conservado el formato original de Matlab®, para facilitar la comprensión de la programación.

```
clear all
close all

directorypath{1}='C:\Users\Edu\Desktop\datos y programas\6-03-2012-4-
m3\06-Mar-2012__14-10-7.dat';

namefile{1}='wave.mat';
namefile{2}='nofil.mat';
namefile{3}='smoothfil.mat';
namefile{4}='hilbertfil.mat';
namefile{5}='waveletfil.mat';

[kk nfile]=size(directorypath);

SUMFFTA=0;
SUMFFTB=0;
SUMFFTC=0;
SUMFFTD=0;

for j=1:nfile

    name=directorypath{nfile};

    x=importdata(name);

    A=x.data; %%señal temporal original
    N=length(A);
    B=smooth(A); % señal temporal filtrada
```

8. Anexos

```
C1=hilbert(A);
C=abs(C1); % señal filtrada con hilbert
D= wden(A, 'sqtwolog', 's', 'one', 3, 'db6'); %señal filtrada con
wavelet

FA=fft(A).*conj(fft(A))./N;
SUMFFTA=FA+SUMFFTA;

FB=fft(B).*conj(fft(B))./N;
SUMFFTB=FB+SUMFFTB;

FC=fft(C).*conj(fft(C))./N;
SUMFFTC=FC+SUMFFTC;

FD=fft(D).*conj(fft(D))./N;
SUMFFTD=FD+SUMFFTD;

end

if (mod(N,2)==0)
    k=0:N/2-1;
else
    k=0:(N-1)/2;
end
T=16384/6000;
freq=(k/T)';

tiempo=(0:1/6000:T-1/6000);

figure();
plot(tiempo,A);%señal original
title('Original signal')
figure();
plot(tiempo,B); %señal filtrada smooth
title('Smooth signal')
figure();
plot(tiempo,C); %señal filtrada hilbert
title('Hilbert signal')
figure();
plot(tiempo,D); %señal filtrada wavelet
title('Wavelet signal')

figure(); %FFT de la señal original
if (mod(N,2)==0)
    plot(freq,FA(1:N/2));
    title('Average FFT before filtering')
else
    plot(freq,FA(1:(N-1)/2));
    title('Average FFT before filtering')
end

figure(); %FFT de la señal filtrada con Smooth
if (mod(N,2)==0)
    plot(freq,FB(1:N/2));
```


8. Anexos

```
        title('Average FFT Smooth filtering')
    else

        plot(freq',FB(1:(N-1)/2));
        title('Average FFT Smooth filtering')
    end

    figure(); %FFT de la señal filtrada con Hilbert
    if (mod(N,2)==0)
        plot(freq,FC(1:N/2));
        title('Average FFT Hilbert filtering')
    else
        plot(freq',FC(1:(N-1)/2));
        title('Average FFT Hilbert filtering')
    end

    figure(); %FFT de la señal filtrada con Wavelet
    if (mod(N,2)==0)
        plot(freq,FD(1:N/2));
        title('Average FFT Wavelet filtering')
    else
        plot(freq',FD(1:(N-1)/2));
        title('Average FFT Wavelet filtering')
    end

    end

FB=FB(1:N/2);

save(namefile{1},'A');
save(namefile{2},'FA');
save(namefile{3},'FB');
save(namefile{4},'FC');
save(namefile{5},'FD');
```

Anexo II: Filtrado de todas las señales de una carpeta con un filtro determinado

Este programa filtra todas las señales de la carpeta especificada mediante los tres tipos distintos de filtrado pero solo guarda las señales procesadas mediante uno de los filtros. Para almacenar todas las señales con todos los filtrados, es necesario lanzar el programa tres veces cambiando cada vez la señal que será guardada, dentro del bucle.

```
clear all
close all

directorypath{1}='C:\Users\Edu\Desktop\datos y programas\datos
eje4\24-04-2012-t\24-04-2012-t-1';

namefile{1}='E3S20hz1s.mat';
namefile{2}='E3S20hz2s.mat';
namefile{3}='E3S20hz3s.mat';
namefile{4}='E3S20hz4s.mat';
namefile{5}='E3S20hz5s.mat';
namefile{6}='E3S20hz6s.mat';
namefile{7}='E3S20hz7s.mat';
namefile{8}='E3S20hz8s.mat';
namefile{9}='E3S20hz9s.mat';
namefile{10}='E3S20hz10s.mat';
namefile{11}='E3S20hz11s.mat';
namefile{12}='E3S20hz12s.mat';
namefile{13}='E3S20hz13s.mat';
namefile{14}='E3S20hz14s.mat';
namefile{15}='E3S20hz15s.mat';
namefile{16}='E3S20hz16s.mat';
namefile{17}='E3S20hz17s.mat';
namefile{18}='E3S20hz18s.mat';
namefile{19}='E3S20hz19s.mat';
namefile{20}='E3S20hz20s.mat';
namefile{21}='E3S20hz21s.mat';
namefile{22}='E3S20hz22s.mat';
namefile{23}='E3S20hz23s.mat';
namefile{24}='E3S20hz24s.mat';
namefile{25}='E3S20hz25s.mat';
namefile{26}='E3S20hz26s.mat';
namefile{27}='E3S20hz27s.mat';
namefile{28}='E3S20hz28s.mat';
namefile{29}='E3S20hz29s.mat';
namefile{30}='E3S20hz30s.mat';
namefile{31}='E3S20hz31s.mat';
namefile{32}='E3S20hz32s.mat';
namefile{33}='E3S20hz33s.mat';
namefile{34}='E3S20hz34s.mat';
namefile{35}='E3S20hz35s.mat';
namefile{36}='E3S20hz36s.mat';
```

8. Anexos

```
namefile{37}='E3S20hz37s.mat';  
namefile{38}='E3S20hz38s.mat';  
namefile{39}='E3S20hz39s.mat';  
namefile{40}='E3S20hz40s.mat';  
namefile{41}='E3S20hz41s.mat';  
namefile{42}='E3S20hz42s.mat';  
namefile{43}='E3S20hz43s.mat';  
namefile{44}='E3S20hz44s.mat';  
namefile{45}='E3S20hz45s.mat';  
namefile{46}='E3S20hz46s.mat';  
namefile{47}='E3S20hz47s.mat';  
namefile{48}='E3S20hz48s.mat';  
namefile{49}='E3S20hz49s.mat';  
namefile{50}='E3S20hz50s.mat';  
namefile{51}='E3S20hz51s.mat';  
namefile{52}='E3S20hz52s.mat';  
namefile{53}='E3S20hz53s.mat';  
namefile{54}='E3S20hz54s.mat';  
namefile{55}='E3S20hz55s.mat';  
namefile{56}='E3S20hz56s.mat';  
namefile{57}='E3S20hz57s.mat';  
namefile{58}='E3S20hz58s.mat';  
namefile{59}='E3S20hz59s.mat';  
namefile{60}='E3S20hz60s.mat';  
namefile{61}='E3S20hz61s.mat';  
namefile{62}='E3S20hz62s.mat';  
namefile{63}='E3S20hz63s.mat';  
namefile{64}='E3S20hz64s.mat';  
namefile{65}='E3S20hz65s.mat';  
namefile{66}='E3S20hz66s.mat';  
namefile{67}='E3S20hz67s.mat';  
namefile{68}='E3S20hz68s.mat';  
namefile{69}='E3S20hz69s.mat';  
namefile{70}='E3S20hz70s.mat';  
namefile{71}='E3S20hz71s.mat';  
namefile{72}='E3S20hz72s.mat';  
namefile{73}='E3S20hz73s.mat';  
namefile{74}='E3S20hz74s.mat';  
namefile{75}='E3S20hz75s.mat';  
namefile{76}='E3S20hz76s.mat';  
namefile{77}='E3S20hz77s.mat';  
namefile{78}='E3S20hz78s.mat';  
namefile{79}='E3S20hz79s.mat';  
namefile{80}='E3S20hz80s.mat';  
namefile{81}='E3S20hz81s.mat';  
namefile{82}='E3S20hz82s.mat';  
namefile{83}='E3S20hz83s.mat';  
namefile{84}='E3S20hz84s.mat';  
namefile{85}='E3S20hz85s.mat';  
namefile{86}='E3S20hz86s.mat';  
namefile{87}='E3S20hz87s.mat';  
namefile{88}='E3S20hz88s.mat';  
namefile{89}='E3S20hz89s.mat';  
namefile{90}='E3S20hz90s.mat';  
namefile{91}='E3S20hz91s.mat';  
namefile{92}='E3S20hz92s.mat';  
namefile{93}='E3S20hz93s.mat';  
namefile{94}='E3S20hz94s.mat';  
namefile{95}='E3S20hz95s.mat';  
namefile{96}='E3S20hz96s.mat';
```

8. Anexos

```
namefile{97}='E3S20hz97s.mat';
namefile{98}='E3S20hz98s.mat';
namefile{99}='E3S20hz99s.mat';
namefile{100}='E3S20hz100s.mat';

[kk nfile]=size(directorypath);

for j=1:nfile
[filenames,amount]=ReadImDir(directorypath{j});
    for i=3:amount
        name=fullfile(directorypath{j},filenames{i});
        x=importdata(name);

        A=x.data;%%señal temporal original
        N=length(A);
        B=smooth(A);% señal temporal filtrada
        C1=hilbert(A);
        C=abs(C1); % señal filtrada con hilbert
        D= wden(A,'sqtwolog','s','one',3,'db6'); %señal filtrada con
        wavelet

        save(namefile{i-2},'D'); %señal que será guardada

    end
end
```

Anexo III: Programa para obtener los histogramas de los valores medios, máximos y mínimos de las señales de una carpeta

Permite obtener los histogramas de la carpeta seleccionada.

```
clear all
close all

directorypath{1}='C:\Users\Edu\Desktop\datos y programas\6-03-2012-4-
m3';

[kk nfile]=size(directorypath);

for j=1:nfile
[filenames,amount]=ReadImDir(directorypath{j});
    for i=3:amount
        name=fullfile(directorypath{j},filenames{i});
        x=importdata(name);

        A=x.data;%%señal temporal original
        N=length(A);
        media(i-2)=mean(A);
        minimo(i-2)=min(A);
        maximo(i-2)=max(A);

    end
end

figure ()
hist(media,100);
figure ()
hist(minimo,100)
figure ()
hist(maximo,100)
```

Anexo IV: Programa para la obtención de los paquetes de energía Wavelet

Este programa obtiene los paquetes de energía de las señales introducidas. Se ha lanzado tres veces, una para cada tipo de filtrado al que han sido sometidas las señales. Para cada lanzamiento es necesario cambiar las señales de entrada y el nombre de las señales de salida. En el caso adjunto, sirve para las señales filtradas con la Transformada Wavelet.

```
clear all
close all
```

```
directorypath{1}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz3';
directorypath{2}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz4';
directorypath{3}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz5';
directorypath{4}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz6';
directorypath{5}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz7';
directorypath{6}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz8';
directorypath{7}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz9';
directorypath{8}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz10';
directorypath{9}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz11';
directorypath{10}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz12';
directorypath{11}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz13';
directorypath{12}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz14';
directorypath{13}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz15';
directorypath{14}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E3S20hz16';
directorypath{15}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E100D20hz3';
directorypath{16}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E100D20hz4';
directorypath{17}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E100D20hz5';
directorypath{18}='C:\Users\Edu\Desktop\datos y
programas\FILTERSNUEVO\temporal wavelet\E100D20hz6';
```

8. Anexos

```
directorypath{19}='C:\Users\Edu\Desktop\datos_y  
programas\FILTERSNUEVO\temporal_wavelet\E100D20hz7';  
directorypath{20}='C:\Users\Edu\Desktop\datos_y  
programas\FILTERSNUEVO\temporal_wavelet\E100D20hz8';  
directorypath{21}='C:\Users\Edu\Desktop\datos_y  
programas\FILTERSNUEVO\temporal_wavelet\E100D20hz9';  
  
namefile{1}='E3S20hz3w.mat';  
namefile{2}='E3S20hz4w.mat';  
namefile{3}='E3S20hz5w.mat';  
namefile{4}='E3S20hz6w.mat';  
namefile{5}='E3S20hz7w.mat';  
namefile{6}='E3S20hz8w.mat';  
namefile{7}='E3S20hz9w.mat';  
namefile{8}='E3S20hz10w.mat';  
namefile{9}='E3S20hz11w.mat';  
namefile{10}='E3S20hz12w.mat';  
namefile{11}='E3S20hz13w.mat';  
namefile{12}='E3S20hz14w.mat';  
namefile{13}='E3S20hz15w.mat';  
namefile{14}='E3S20hz16w.mat';  
namefile{15}='E100D20hz3w.mat';  
namefile{16}='E100D20hz4w.mat';  
namefile{17}='E100D20hz5w.mat';  
namefile{18}='E100D20hz6w.mat';  
namefile{19}='E100D20hz7w.mat';  
namefile{20}='E100D20hz8w.mat';  
namefile{21}='E100D20hz9w.mat';  
  
des=3; %nivel de descomposicion  
[kk nfile]=size(namefile);  
  
for j=1:nfile  
    j  
    [filenames,amount]=ReadImDir(directorypath{j});  
    for i=3:amount  
        name=fullfile(directorypath{j},filenames{i});  
        Estr(:,i-2)=load(name);  
        X(:,i-2)=Estr(:,i-2).D;  
    end  
    [ndata,nfile]=size(X);  
  
    % Wavelet packet para el nivel de descomposicion des  
    for i=1:nfile  
        T(:,i)=wpdec(X(:,i),des,'db6');  
        E(:,i)=wenergy(T(:,i));  
    end  
    save(namefile{j},'E');  
end
```

Anexo V: Programa para crear la red neuronal a partir de todas las carpetas de señales (caso 1)

Este programa crea la red neuronal a partir de todas las señales disponibles sometidas al tipo de filtrado seleccionado. En el caso del programa adjunto, se crea a partir de las señales tratadas mediante el filtro Wavelet. Para cada tipo de filtro se van cambiando las entradas.

Este programa corresponde al caso 1 de los estudiados en el capítulo de redes neuronales.

```
clear all
close all

%cargar ficheros para entrenamiento (almacenamos los nombres (o ruta)
de los ficheros a cargar en namefile{},
%almacenamos si el eje es sano o con defecto en targetfile{})

namefile{1}='E3S20hz3w.mat';
targetfile{1}=zeros(1,100);

namefile{2}='E3S20hz4w.mat';
targetfile{2}=zeros(1,100);

namefile{3}='E3S20hz5w.mat';
targetfile{3}=zeros(1,100);

namefile{4}='E3S20hz6w.mat';
targetfile{4}=zeros(1,100);

namefile{5}='E3S20hz7w.mat';
targetfile{5}=zeros(1,100);

namefile{6}='E3S20hz8w.mat';
targetfile{6}=zeros(1,100);

namefile{7}='E3S20hz9w.mat';
targetfile{7}=zeros(1,100);

namefile{8}='E3S20hz10w.mat';
targetfile{8}=zeros(1,100);

namefile{9}='E100D20hz3w.mat';
targetfile{9}=ones(1,100);

namefile{10}='E100D20hz4w.mat';
targetfile{10}=ones(1,100);
```


8. Anexos

```
namefile{11}='E100D20hz5w.mat';
targetfile{11}=ones(1,100);

namefile{12}='E100D20hz6w.mat';
targetfile{12}=ones(1,100);

namefile{13}='E3S20hz11w.mat';
targetfile{13}=zeros(1,100);

namefile{14}='E3S20hz12w.mat';
targetfile{14}=zeros(1,100);

namefile{15}='E3S20hz13w.mat';
targetfile{15}=zeros(1,100);

namefile{16}='E3S20hz14w.mat';
targetfile{16}=zeros(1,100);

namefile{17}='E3S20hz15w.mat';
targetfile{17}=zeros(1,100);

namefile{18}='E3S20hz16w.mat';
targetfile{18}=zeros(1,100);

namefile{19}='E100D20hz7w.mat';
targetfile{19}=ones(1,100);

namefile{20}='E100D20hz8w.mat';
targetfile{20}=ones(1,100);

namefile{21}='E100D20hz9w.mat';
targetfile{21}=ones(1,100);

[kk nfiles]=size(targetfile);
for i=1:nfiles

    load(namefile{i});
    if i==1
        P=E;
        T=targetfile{i};
    else
        P=cat(2,P,E);
        T=cat(2,T,targetfile{i});
    end
end
```

8. Anexos

```
% CREACIÓN Y ENTRENAMIENTO DE UNA RED DE RECONOCIMIENTO DE PATRONES
netclas=newpr(P,T,8);
netclas.trainParam.goal = 0;
netclas.trainParam.epochs = 10000;
netclas.trainParam.max_fail = 100;

% netclas=newpr(P,T,no);
% P: patrones de entrenamiento
% T: target, salida conocida
% no: número de neuronas en la capa oculta

netclas = train(netclas,P,T);
```

Anexo VI: Programa para crear la red neuronal a partir de unas señales y testear con otras diferentes (caso 2)

Este programa crea la red neuronal a partir de las señales de unas carpetas y testea a partir de las señales de otras carpetas distintas. En el caso del programa adjunto, se crea a partir de las señales tratadas mediante el filtro Wavelet. Para cada tipo de filtro se van cambiando las entradas.

Este programa corresponde al caso 2 de los estudiados en el capítulo de redes neuronales.

```
clear all
close all

%cargar ficheros para entrenamiento (almacenamos los nombres (o ruta)
de los ficheros a cargar en namefile{},
%almacenamos si el eje es sano o con defecto en targetfile{})

namefile{1}='E3S20hz3w.mat';
targetfile{1}=zeros(1,100);

namefile{2}='E3S20hz4w.mat';
targetfile{2}=zeros(1,100);

namefile{3}='E3S20hz5w.mat';
targetfile{3}=zeros(1,100);

namefile{4}='E3S20hz6w.mat';
targetfile{4}=zeros(1,100);

namefile{5}='E3S20hz7w.mat';
targetfile{5}=zeros(1,100);

namefile{6}='E3S20hz8w.mat';
targetfile{6}=zeros(1,100);

namefile{7}='E3S20hz9w.mat';
targetfile{7}=zeros(1,100);

namefile{8}='E3S20hz10w.mat';
targetfile{8}=zeros(1,100);

namefile{9}='E100D20hz3w.mat';
targetfile{9}=ones(1,100);
```

8. Anexos

```
namefile{10}='E100D20hz4w.mat';
targetfile{10}=ones(1,100);

namefile{11}='E100D20hz5w.mat';
targetfile{11}=ones(1,100);

namefile{12}='E100D20hz6w.mat';
targetfile{12}=ones(1,100);

[kk nfiles]=size(targetfile);
for i=1:nfiles

    load(namefile{i});
    if i==1
        P=E;
        T=targetfile{i};
    else
        P=cat(2,P,E);
        T=cat(2,T,targetfile{i});
    end

end

% CREACIÓN Y ENTRENAMIENTO DE UNA RED DE RECONOCIMIENTO DE PATRONES
netclas=newpr(P,T,8);
netclas.trainParam.goal = 0;
netclas.trainParam.epochs = 10000;
netclas.trainParam.max_fail = 100;
netclas.divideParam.trainRatio = 0.98;
netclas.divideParam.valRatio = 0.01;
netclas.divideParam.testRatio = 0.01;
% netclas=newpr(P,T,no);
% P: patrones de entrenamiento
% T: target, salida conocida
% no: número de neuronas en la capa oculta

netclas = train(netclas,P,T);

% PRUEBA DE LA RED CON PATRONES DIFERENTES

namefile1{1}='E3S20hz11w.mat';
Tval=zeros(1,100);
namefile1{2}='E3S20hz12w.mat';
Tval=[Tval zeros(1,100)];
namefile1{3}='E3S20hz13w.mat';
Tval=[Tval zeros(1,100)];
namefile1{4}='E3S20hz14w.mat';
Tval=[Tval zeros(1,100)];
namefile1{5}='E3S20hz15w.mat';
Tval=[Tval zeros(1,100)];
namefile1{6}='E3S20hz16w.mat';
Tval=[Tval zeros(1,100)];

namefile1{7}='E100D20hz7w.mat';
Tval=[Tval ones(1,100)];
```

8. Anexos

```
namefile1{8}='E100D20hz8w.mat';
Tval=[Tval ones(1,100)];
namefile1{9}='E100D20hz9w.mat';
Tval=[Tval ones(1,100)];

[kk nfiles]=size(namefile1);
for i=1:nfiles
    load(namefile1{i});
    % E=mean(E,2);
    if i==1
        patron=E;
    else
        patron=cat(2,patron,E);
    end
end

A=sim(netclas,patron);

%Si A<0.5, A=0,
%Si A>=0.5; A=1;
tasadeacierto=0;
for i=1:900
    if A(i)<0.5;
        A(i)=0;
    else
        A(i)=1;
    end

    if Tval(i)==A(i);
        tasadeacierto=tasadeacierto + 1;
    end
end
%Calcular tasa de acierto
tasadeacierto=tasadeacierto/900*100

[x,y]=size(A);
plot(A,'b');
axis([0 y -0.2 1.2])
hold on
plot(Tval,'r')
```